

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації та управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

на тему: «Інформаційна система підтримки діяльності
фалеристів»

Виконав:

студент IV курсу, групи ІС-62

Ковинєв Кирило Олексійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

Доцент, к.т.н. Жураковська Оксана Сергіївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

Доцент, к.т.н. Новінський Валерій Петрович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

Доцент, к.т.н. Лісовиченко Олег Іванович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

(підпис)

Київ – 2020 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ” 2020 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Ковинєву Кирилу Олексійовичу

(прізвище, ім'я, по батькові)

1. Тема проєкту «Інформаційна система підтримки діяльності фалеристів»

керівник проєкту Жураковська Оксана Сергіївна, к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7”травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01”червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. *Схема структурна варіантів використань*

2. *Схема бази даних*

3. *Схема структурна роботи алгоритму*

4. *Схема структурна класів програмного забезпечення*

5. *Схема структурна компонентів програмного забезпечення*

6. *Схема структурна послідовності*

7. *Креслення вигляду екранних форм*

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>14.04.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>15.04.2020</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>16.04.2020</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>18.04.2020</i>	
5.	<i>Алгоритмізація задачі</i>	<i>20.04.2020</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>23.04.2020</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>05.05.2020</i>	
8.	<i>Налагодження програми</i>	<i>07.05.2020</i>	
9.	<i>Виконання графічних документів</i>	<i>10.05.2020</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>14.05.2020</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>15.05.2020</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
13.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Кирило КОВИНЄВ

Керівник

Оксана ЖУРАКОВСЬКА

[illegible]

**Пояснювальна записка
до дипломного проєкту**

на тему: Інформаційна система підтримки діяльності фалеристів

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи: пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 14 рисунків, 21 таблицю, 1 додаток, 10 джерел.

Дипломний проєкт присвячений розробці системі підтримки діяльності роботи фалеристів. Головною ціллю є спрощення та підвищення ефективності процесу колекціонування та експонування за рахунок розміщення та зберігання електронної версії колекції на сайті.

У розділі інформаційного забезпечення вказано вхідні дані для системи, вихідні дані та опис структури бази даних, яку використовує проєкт.

Розділ математичного забезпечення присвячений задачі, яку потрібно вирішити для функціонування системи, її змістовну постановку, її математичну постановку. Також наведені методи розв'язання, їх аналіз та вибір оптимального методу.

Програмне забезпечення містить у собі опис засобів, які використовувалися у розробці системи, опис вимог до технічного забезпечення, опис архітектури системи та специфікація функцій.

У технологічному розділі наведено керівництво користувача з детальними інструкціями та опис проведених тестів, результати яких показують на працездатність системи.

Ключові слова: МЕДАЛЬ, КОЛЕКЦІЯ, КАТАЛОГ МЕДАЛЕЙ, КОЛЕКЦІОНЕРИ НАСТІЛЬНИХ МЕДАЛЕЙ

					ДП 6210 00.000 ПЗ			
Зм.	Арк.	ПІБ	Підп.	Дата	Пояснювальна записка	Лит.	Арк.	Аркушів
Розробн.		Ковинев К.О.						
Керівн.		Жураковська О.С.					2	89
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
Н/контр.		Новінський В.П.						
В.о.зав.каф.		Павлов О.А.						

ABSTRACT

Structure and scope of work: the explanatory note of the diploma consists of five sections and contains 14 figures, 21 tables, 1 appendix, and 10 sources.

The diploma is devoted to the developing a support system for phalerist activity. The main goal is to simplify and increase the efficiency of the process of collecting and exhibiting by placing and storing an electronic version of the collection on a site.

The information section provides input for the system, output, and a description of the database structure used by the project.

The section of mathematical support is devoted to the problem that needs to be solved for the functioning the system, its meaningful formulation, and its mathematical formulation. Methods of solution, their analysis and selection of the optimal method are also given.

The software includes a description of the tools used in the development of the system, a description of the hardware requirements, a description of the system architecture, and a specification of the functions.

The technological section provides the user manual with detailed instructions and a description of the tests performed, the results of which show the efficiency of the system.

MEDAL, COLLECTION, CATALOG OF MEDALS, COLLECTORS OF
TABLE MEDALS

					ДП ІС-6210.00.000 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

Вступ.....	6
1. ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	8
1.1 Опис предметного середовища	8
1.2 Огляд аналогів існуючих рішень.....	9
1.3 Опис функціональної моделі.....	10
1.4 Постановка задачі.....	11
1.4.1 Призначення розробки	11
1.4.2 Цілі та задачі розробки.....	12
Висновок до розділу	13
2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	14
2.1 Вхідні дані	14
2.2 Вихідні дані.....	15
2.3 Опис структури бази даних	15
Висновок до розділу	18
3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	19
3.1 Змістовна постановка задачі	19
3.2 Математична постановка задачі	19
3.3 Обґрунтування методу розв'язання	19
3.4 Опис методів розв'язання.....	20
Висновок до розділу	21
4. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	22
4.1 Засоби розробки	22
4.2 Вимоги до технічного забезпечення	24
4.3 Архітектура програмного забезпечення	24
4.4 Специфікація функцій	26
Висновок до розділу	32
5. ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	33
5.1 Керівництво користувача.....	33

5.1.1	Опис головного каталогу	33
5.1.2	Опис авторизації.....	36
5.1.3	Опис реєстрації.....	36
5.1.4	Опис власної колекції.....	37
5.1.5	Опис додання медалі.....	39
5.1.6	Опис зміни інформації про медаль	40
5.1.7	Опис користувацького каталогу.....	40
5.1.8	Опис списку діалогів.....	41
5.1.9	Опис діалогу.....	42
5.1.10	Опис інформації про медаль	43
5.1.11	Опис інформації про користувача	44
5.2	Випробування програмного продукту	45
5.2.1	Мета випробувань.....	45
5.2.2	Загальні положення	45
5.2.3	Результати випробувань	46
	Висновок до розділу	54
	Загальні висновки	55
	Перелік посилань	57
	ДОДАТОК А.....	58

ВСТУП

У наш час існує багато різноманітних занять або хобі. До них належить колекціонування. У його основі лежить збір предметів схожих за типом, параметрами або характеристиками. Ці ж предмети можуть мати історичну, наукову або художню цінність. Колекціонування відрізняється від простого збирання тим, що включає у себе виявлення, збір, вивчення та систематизацію предметів або матеріалів. Об'єктом колекціонування може бути будь-яка річ: починаючи від предметів історичної пам'ятки (стародавні монети, книги, витвори образотворчого мистецтва відомих особистостей, рукописи тощо) та закінчуючи об'єктами природи (цікаві рослини, комахи або мінерали з різних куточків світу). У суспільстві колекціонування вважається своєрідним прикладом творчості, а будь-яка колекція є результатом цієї творчої діяльності.

Фалеристика — дисципліна, що вивчає історію створення та існування орденів, значків, медалей, та будь-яких нагрудних або настільних знаків. Фалеристикою також називають і колекціонування цих предметів.

Одним з прикладів колекціонування серед фалеристів є колекціонування настільних медалей. Історія цих медалей починається в XVI столітті. Майстра почали створювати новий предмет розкоші, показуючи свої уміння та навички в художньому мистецтві. Саме цей час можна назвати початком існування сувенірних медалей. Їх призначенням було стати прикрасою або пам'ятним подарунком. У наш час медалі популярні і досі. Яскравим прикладом є настільні медалі, що випускаються до якої-небудь круглої дати або пам'ятної події. Приводом для випуску може бути діяльність відомої особистості, що залишила вагомий слід у житті суспільства [1].

Однією з головних проблем колекціонування є експонування. Для багатьох колекціонерів демонстрація є важливою частиною їх

					ДП ІС-6210.00.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

колекціонування, адже завжди приємно продемонструвати результати своєї творчої діяльності. Популярним рішенням цього питання є організація виставок або розміщення електронного варіанту колекції в Інтернеті. Також є випадки, коли власники або їх нащадки віддають різні приватні колекції до музеїв.

Другою, але не менш важливою, є проблема зберігання інформації про власну колекцію. Основними характеристиками настільної медалі є: країна випуску, рік випуску, матеріал, діаметр, вага, тема. І дуже часто виникає потреба дізнатися скільки медалей того чи іншого року знаходиться у колекції. Без зберігання інформації про колекцію, рішення потреби може витратити багато сил та часу. Особливо коли колекція налічує сотні або навіть тисячі екземплярів.

Веб-застосування, що розробляється в даному дипломному проєкті, повинно забезпечити колекціонерам зручне зберігання інформації про власну колекції та простий доступ до неї. Також система повинна надавати можливість демонстрації електронних версій різних колекцій настільних медалей іншим колекціонерам.

					ДП ІС-6210.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Система повинна створюватися з урахуванням наступних особливостей: тільки авторизований колекціонер може вести та демонструвати свою електронну версію колекцію; переглядати каталоги з настільними медалями може будь-який користувач; колекціонер може мати декілька настільних медалей; настільна медаль може бути у декількох колекціонерів; не уся інформація про характеристики медалі може бути відома.

Кожна медаль характеризується наступними параметрами: назва, країна випуску, рік випуску, матеріал, діаметр, вага, опис, фото. Для медалей потрібно зберігати інформацію про користувачів, в яких вона є.

Кожен зареєстрований користувач характеризується наступними параметрами: прізвище, ім'я, логін, електронна пошта, пароль. Для користувачів потрібно зберігати інформацію про медалі, які у нього є.

Система створюється для наступних груп користувачів: адміністрація, модератори, звичайні користувачі.

Множина медалей називається «каталогом». Множина електронних версій медалей користувача, які є в його колекції, називаються «каталогом власної колекції». Множина медалей електронних версій, які розміщені на сайті модераторами, називаються «головним\основним каталогом сайту».

Користувачі повинні мати можливість інформаційної підтримки бази даних застосунку: ведення бази даних (запис, читання, модифікація, видалення), забезпечення логічної несуперечливості бази даних.

Користувачі застосунку повинні мати можливість: ведення каталогу власної колекції, отримати список медалей конкретного каталогу власної колекції користувача або головного каталогу сайту, отримати інформацію

					ДП ІС-6210.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

про каталог, отримати інформацію про користувача, отримати інформацію про конкретну медаль, комунікації з іншими користувачами.

Модератори та адміністрація повинні підтримувати контент сайту у належному вигляді та поповнювати головний каталог сайту.

Модератори застосунку повинні мати можливість: видалення медалі з будь-якого каталогу на сайті, ведення головного каталогу сайту.

Адміністратори застосунку повинні мати можливість: видалення медалі з будь-якого каталогу на сайті, ведення головного каталогу сайту, зміна ролі/статусу користувача.

Предметом комунікації між користувачами виступає чат з текстовими повідомленнями, кожне з яких має наступні характеристики: зміст повідомлення, час відправлення, відправник, отримувач.

1.2 Огляд аналогів існуючих рішень

Розглянемо існуючі сайти для зберігання та демонстрації медалей:

1) Веб-сайт Collector Coins

Переваги:

- великий каталог медалей з усього світу;
- зручний інтерфейс;
- можливість пошуку медалі за допомогою фільтрів.

Недоліки:

- немає можливості ведення власної колекції.

2) Веб-сайт Colnect

Переваги:

- доступний на 50 мовах;
- наявність багатьох характеристик предмету колекціонування, для більш точного пошуку.

Недоліки:

- незручний інтерфейс;

					ДП ІС-6210.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- сайт призначений не для одного конкретного предмету колекціонування.

3) Веб-сайт Lagid

Переваги:

- доступний на трьох мовах;
- можливість комунікації з адміністрацією для покращення роботи сайту.

Недоліки:

- незручний інтерфейс;
- сайт призначений не для одного конкретного предмету колекціонування.

В результаті аналізу існуючих рішень, можна зробити висновок, що є актуальність розробки системи підтримки діяльності фалеристів з урахуванням усіх недоліків аналогів та їх усуненням. Система повинна:

- мати зручний інтуїтивно зрозумілий інтерфейс;
- бути призначеною для одного конкретного предмету колекціонування;
- мати можливість ведення власної колекції.

1.3 Опис функціональної моделі

Дійові особи, які будуть працювати в межах розробленого програмного продукту:

- Користувач – основною метою є : організація процесу колекціонування та розширення можливостей доступу до інформації та обміну інформацією для колекціонування.
- Модератор – основною метою є підтримання контенту сайту.
- Адміністратор – основною метою є підтримка коректної роботи сайту.

					ДП ІС-6210.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Функції, які будуть виконуватися дійовими особами наведено у структурній схемі діяльності у графічній частині матеріалу.

Користувач:

- реєстрація\авторизація;
- ведення каталогу власної колекції;
- перегляд інформації про власну колекцію;
- перегляд інформації про іншого користувача та його колекцію;
- перегляд головного каталогу з медалями сайту, використовуючи фільтри характеристик предмету колекціонування;
- перегляд колекцій інших користувачів, використовуючи фільтри характеристик предмету колекціонування;
- пошук конкретної медалі;
- пошук користувача за ім'ям;
- комунікація з іншим користувачем за допомогою повідомлень;
- додання медалі до каталогу власної колекції з головного каталогу сайту.

Модератор:

- видалення предмету колекціонування з каталогу власної колекції іншого користувача;
- ведення основного каталогу сайту.

Адміністратор:

- зміна статусу користувачів та модераторів.

1.4 Постановка задачі

1.4.1 Призначення розробки

Призначенням розробки застосування є підтримка діяльності колекціонерів настільних медалей

					ДП ІС-6210.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1.4.2 Цілі та задачі розробки

Цілі розробки:

Спрощення та підвищення ефективності процесу колекціонування та експонування за рахунок розміщення та зберігання електронної версії колекції на сайті.

Задачі розробки:

- проектування та створення бази даних;
- реалізація можливості реєстрації та авторизації користувачу;
- організація можливості ведення каталогу власної колекції користувачу та головного каталогу сайту модератору та адміністратору;
- організація перегляду головного каталогу сайту та каталогів колекцій інших користувачів;
- реалізація перегляду інформації про конкретну медаль;
- реалізація використання фільтрів до каталогів за характеристиками предмету колекціонування;
- організація чату між користувачами;
- надання можливості адміністратору зміни статусу користувачів;
- реалізація зручного пошуку медалі серед головного каталогу сайту та каталогів користувачів;
- реалізація пошуку конкретного користувача за логіном;
- організація додання предмету колекціонування користувачем з головного каталогу сайту до каталогу власної колекції;
- організація додання предмету колекціонування модератором або адміністратором з каталогу власної колекції іншого користувача до головного каталогу сайту.

					ДП ІС-6210.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Висновок до розділу

У даному розділі було розглянуто цілі та головні задачі розробки системи, опис предметного середовища, огляд аналогів існуючих рішень та опис функціональної моделі. Було наведено інформацію про особливості колекціонування настільних медалей та їх параметри.

Дана робота присвячена підтримки діяльності колекціонерів настільних медалей, а саме спрощенню процесу експонування за рахунок розміщення та зберігання колекції на сайті.

Було розглянуто аналоги існуючих рішень, виявлено їх недоліки та переваги. З урахуванням недоліків було зроблено висновок про необхідність та актуальність розробки системи.

У розділі було надано опис дійових осіб та їх головні призначення. За допомогою структурної схеми використань можна побачити функції, які будуть виконуватися дійовими особами.

Був проведений аналіз потреб колекціонерів для формування конкретних задач, які необхідно вирішити для розробки застосування.

					ДП ІС-6210.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані для задачі підтримки діяльності фалеристів можна поділити на наступні категорії:

1) Вхідні дані для наповненості сайту: Система надає можливість перегляду інформації та зображення медалі з головного каталог. Також система надає можливість користувачу ведення каталогу власної колекції за допомогою додання медалей з головного каталогу сайту. Тому для цього треба наповнити головний каталог інформацією та зображеннями про медалі у кількості достатньої для зручного користування перших користувачів (близько 300 медалей). Інформація на кожну медаль: назва, країна випуску, рік випуску, категорія, матеріал, вага, діаметр, опис, зображення.

2) Вхідні дані першого користувача зі статусом «Адміністратор»: Для підтримки контенту сайту є користувачі зі статусом «Модератор» та «Адміністратор». Вони з'являються шляхом зміни статусу звичайному користувачу адміністратором. Так як система не передбачає самостійної зміни статусу користувачем або при реєстрації, з початку функціонування системи потрібен хоча б один адміністратор. Інформація про адміністратора: ім'я, прізвище, логін, пошта, пароль, статус «Адміністратор».

3) Вхідні дані користувача, діяльність якого буде підтримувати система:

Для ведення каталогу користувачем, система потребує реєстрації користувача для збереження даних за якими буде закріплюватися власний каталог.

Інформація про адміністратора: ім'я, прізвище, логін, пошта, пароль.

					ДП ІС-6210.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Вихідні дані

Вихідними даними задачі буде інформація про кожну окрему медаль: назва, країна випуску, рік випуску, категорія, матеріал, вага, діаметр, опис, зображення.

Сукупність медалей утворює каталоги: каталог власної колекції користувача, користувацький каталог, головний каталог сайту.

Також медалі утворюють список результатів за запитом користувача: пошук медалі за назвою, пошук за допомогою фільтрів, сортування за параметром.

2.3 Опис структури бази даних

Схему структури бази даних наведено у графічній частині матеріалу.

Розглянемо призначення кожної таблиці:

- Users – таблиця з інформацією про кожного користувача;
- Products – таблиця з інформацією про кожну медаль;
- UserProduct – таблиця для забезпечення зв'язку багато до багатьох таблиць Users та Products;
- Dialogues – містить унікальний номер кожного діалогу між користувачами;
- UserDialogue – таблиця для забезпечення зв'язку багато до багатьох таблиць Users та Dialogues;
- Messages – таблиця з інформацією про кожне повідомлення.

Таблиця 2.1 – Опис таблиці Products

Код	Опис	Тип даних	Первинний	Обов'язкове
ProductId	Унікальний ідентифікатор запису у таблиці	int	X	X
Name	Назва медалі	nvarchar		X
Country	Країна випуску медалі	nvarchar		X
Year	Рік випуску медалі	int		X
Diameter	Діаметр медалі	real		X
Weight	Вага медалі	real		X
Description	Опис медалі	nvarchar		X
Photo	Зображення медалі	varbinary		X
IsMainCatalog	Показник наявності медалі у головному каталозі	bit		X
UserAdderUserId	Ідентифікатор користувача, що додав медаль до сайту	int		X
Material	Матеріал медалі	nvarchar		X
Category	Категорія медалі	nvarchar		X
Time	Час додання медалі до сайту	datetime		X
MinPhoto	Зменшене зображення медалі	varbinary		X

Таблиця 2.2 – Опис таблиці Users

Код	Опис	Тип даних	Первинний	Обов'язкове
UserId	Унікальний ідентифікатор запису у таблиці	int	X	X
Name	Ім'я користувача	nvarchar		X
Surname	Прізвище користувача	nvarchar		X
Login	Логін користувача	nvarchar		X
Email	Електронна пошта користувача	nvarchar		X
Pass	Пароль користувача	nvarchar		X
Role	Статус користувача	nvarchar		X

Таблиця 2.3 – Опис таблиці Dialogues

Код	Опис	Тип даних	Первинний	Обов'язкове
DialogueId	Унікальний ідентифікатор запису у таблиці	int	X	X

Таблиця 2.4 – Опис таблиці Messages

Код	Опис	Тип даних	Первинний	Обов'язкове
MessageId	Унікальний ідентифікатор запису у таблиці	int	X	X
Text	Текст повідомлення	nvarchar		X

Продовження таблиці 2.4

Код	Опис	Тип даних	Первинний	Обов'язкове
Time	Час відправлення повідомлення	datetime		X
DialogueId	Ідентифікатор діалогу, якому належить повідомлення	int		X
SenderId	Ідентифікатор користувача, що відправив повідомлення	int		X

Висновок до розділу

У розділі інформаційного забезпечення було описано вхідні дані для виконання задачі підтримки діяльності фалеристів, а саме: дані для наповненості сайту, дані для першого користувача зі статусом «Адміністратор» та дані користувача, діяльність якого буде підтримувати система.

Вихідними даними є результати запитів користувача за пошуком, фільтрацією та сортуванням, та контент каталогів сайту.

У розділі було описано структуру бази даних та призначення кожної таблиці. Також було наведено опис кожної таблиці та опис кожного стовбця.

3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Заданий список назв медалей. Користувач вводить запит на пошук медалі за назвою. Треба знайти усі медалі, де назва буде мати спільний підрядок з запитом, та відсортувати у порядку зменшення розміру підрядку. Розмір підрядку має бути не менше ніж 40% від розміру запиту. Для усіх назв та запиту відомі довжина та символи.

3.2 Математична постановка задачі

Нехай S_1, S_2, \dots, S_n – назви медалей, де n – кількість назв; M – рядок запиту користувача, m – довжина строки запиту користувача.

Треба знайти підрядки W_1, W_2, \dots, W_n ,

$$W_i = \begin{cases} \emptyset, & \text{якщо } \left(\frac{w_i}{m} * 100\%\right) < 40\%; \\ \max\{W \mid W \subseteq S_i, W \subseteq M\}, & \text{інакше.} \end{cases}$$

де w_i – довжина i -ого підрядку, $i \in (1, n)$.

3.3 Обґрунтування методу розв'язання

Метод перебору

Дозволяє знайти підрядок максимальної довжини двох рядків за допомогою порівняння усіх існуючих підрядків.

Переваги:

- дає точний результат пошуку найбільшого підрядку двох рядків;
- простий у реалізації;
- не потребує додаткових технологій для використання.

Недоліки:

- у найгіршому випадку час роботи алгоритму буде дуже довгим.

Метод пошуку найбільшої підпоследовності [2]

Дозволяє знайти підпоследовність максимальної довжини двох рядків.

Переваги:

- простий у реалізації;
- не потребує додаткових технологій для використання.

Недоліки:

- коли підрядок завжди є підпоследовністю рядка, то підпоследовність не завжди є підрядком рядка. Тому ми можемо отримати не точний результат, а саме знайти підпоследовність, що не є підрядком.

Метод пошуку найбільшого підрядку [3]

Дозволяє знайти підрядок максимальної довжини двох рядків.

Переваги:

- дає точний результат пошуку найбільшої підрядку двох рядків;
- простий у реалізації;
- не потребує додаткових технологій для використання;
- час роботи алгоритму не залежить від найгіршого випадку.

У результаті аналізу трьох алгоритмів, можна зробити висновок, що метод пошуку найбільшого підрядку є самим ефективним у вирішенні поставленої задачі: він немає недоліків у порівнянні з іншими методами, дає точний результат та простий у реалізації.

3.4 Опис методів розв'язання

Для вирішення задачі використовуємо алгоритм пошуку найбільшого спільного підрядку:

Рішення завдання пошуку найбільшого загального підрядку для двох рядків S_1 та S_2 , довжини яких m та n відповідно, полягає в заповненні таблиці A_{ij} розміром $(m+1) \times (n+1)$ за таким правилом, приймаючи, що символи в рядку нумеруються від одиниці:

					ДП ІС-6210.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\begin{cases} A_{0j} = 0, & j = 0 \dots n, \\ A_{i0} = 0, & i = 0 \dots m, \\ A_{ij} = 0, S_1[i] \neq S_2[j], i \neq 0, j \neq 0 \\ A_{ij} = A_{i-1, j-1} + 1, S_1[i] = S_2[j], i \neq 0, j \neq 0 \end{cases}$$

Максимальне число у таблиці це і є довжина найбільшої спільного підрядка.

Схему роботи алгоритму наведено у графічній частині матеріалу.

Висновок до розділу

В розділі математичного забезпечення наведено змістовну та математичну постановку задачі пошуку назви наближеної до запиту користувача. Було наведено три алгоритми для вирішення поставленої задачі. За результатами аналізу був вибраний метод пошуку найбільшого підрядка двох рядків. Наведено детальний опис алгоритму та структурну схему його роботи для розв'язання задачі.

4. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для розробки системи підтримки діяльності роботи колекціонерів було використано мову програмування С#, технологію ASP NET .Core MVC, фреймворк Entity Framework, мову тегів HTML, мову стилю сторінок CSS, та СУБД Microsoft SQL Server для збереження даних.

С# є основною мовою розробки програм на платформі .NET корпорації Microsoft. У ньому вдало поєднуються випробувані засоби програмування з самими останніми нововведеннями і надається можливість для ефективного і дуже практичного написання програм, призначених для обчислювального середовища сучасних підприємств. У мови програмування С# «багата спадщина». Вона є прямим спадкоємцем двох найбільш вдалих мов програмування: С і С++. Вона також має тісні родинні зв'язки з ще однією мовою програмування: Java [4].

ASP.NET Core MVC - це інфраструктура для розробки веб-додатків виробництва Microsoft. яка поєднує в собі ефективність і акуратність архітектури "модель-представлення-контролер" (model-view-controller - MVC), ідеї і прийоми гнучкої розробки. а також кращі частини платформи .NET. Інфраструктура ASP.NET Core MVC пропонує функціональність початкової інфраструктури ASP.NET MVC Framework, побудованої поверх нової платформи ASP.NET Core. Вона інтегрує функціональні засоби, які раніше надавалися Web API, підтримує більш природний спосіб генерування складного вмісту і робить основні завдання розробки, такі як модульне тестування. простішими і передбачуваними [5].

Entity Framework - це інфраструктура об'єктно-реляційного відображення, яка пропонує абстракцію ADO.NET для отримання об'єктної моделі на основі довідкових баз даних. Разом з Entity Framework можна

					ДП ІС-6210.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

використовувати різні моделі програмування: "спочатку модель" (Model First), "спочатку база даних" (Database First) і "спочатку код" (Code First). Моделі "спочатку модель" і "спочатку база даних" надають інформацію відображення у файлі відображення. У разі моделі "спочатку код" все відображення робиться в коді C #. Інфраструктура ADO.NET Entity Framework підтримує пряме визначення сутнісних класів, які абсолютно незалежні від структури бази даних, і відображає їх на таблиці і асоціації бази даних. За рахунок застосування об'єктів додаток захищається від змін, що вносяться до бази даних. Для визначення сутнісно-орієнтованих запитів до сховища в ADO.NET Entity Framework використовується Entity SQL. Технологія LINQ to Entities дозволяє застосовувати синтаксис LINQ для запиту даних. Об'єктний контекст виступає в ролі моста до змінених сутностей, зберігаючи інформацію на випадок, коли сутності повинні записуватися назад в сховище [6].

Microsoft SQL Server – система управління базами даних, розроблена Microsoft. Її головними задачами є збереження даних, управління даними від імені користувача, регламентування складу даних, які можуть бути введені у систему, а також спрощення читання даних із системи. Також СУБД SQL Server надає у розпорядження користувача додаткові служби, які дозволяють автоматизувати взаємодію збережених у ній даних з даними з інших систем завдяки таким потужним засобам, як SQL Server Agent, Integration Services, Notification Services [7].

HTML – це мова розмітки, якою пишуться гіпертекстові документи, за допомогою тегів [8]. Ця мова робить акцент на спрощенні розмітки, необхідної для створення відповідних W3C-стандартам сторінок та об'єднання усього потрібного: CSS- та JavaScript- коду, а також файлів зображень. HTML-документ складається з групи елементів, кожен з яких визначається певним тегом [9].

CSS – спеціальна мова, яка використовується для оформлення сторінок, що були написані мовами розмітки даних. Каскадні таблиці стилів були зроблені у якості способу розподілу дизайну та вмісту [9].

4.2 Вимоги до технічного забезпечення

Мінімальні системні вимоги:

- а) Процесор: з частотою не менше 1 ГГц;
- б) Обсяг ОЗУ: не менше 64 мегабайт;
- в) Операційна система: Windows XP/7/8/10;
- г) Наявність одного з браузерів:
 - 1) Internet Explorer;
 - 2) Opera;
 - 3) Google Chrome;
 - 4) Mozilla Firefox;
 - 5) Microsoft Edge.

4.3 Архітектура програмного забезпечення

Система підтримки діяльності колекціонерів використовує архітектурний шаблон проектування програмного забезпечення "модель-представлення-контролер" (model-view-controller - MVC). У рамках цього шаблону система ділиться на три частини, кожна з яких відповідає за окремі функції:

- модель відповідає за збереження даних та їх структуру;
- представлення бере на себе задачу введення та виведення інформації даних користувачу;
- контролер отримує вхідні дані від користувача та передає дані у модель або у представлення.

Взаємодія користувача з застосуванням, яке використовує патерн MVC, прямує за загальним циклом: користувач виконує дію, а застосунок змінює

свою модель даних у відповідь на дії користувача, та виводить оновлені дані користувачу. Потім цикл повторюється.

Для збереження інформації та даних від користувача до бази даних, був використаний фреймворк Entity Framework. Він полегшує роботу з базою даних та надає можливість зручного маніпулювання даними. У частині графічного матеріалу наведена схема класів сутностей системи. Вона відображає моделі та залежності між ними.

Для демонстрації архітектури застосунку використовується діаграма компонентів, що наведена у частині графічного матеріалу. Кожний компонент є простором імен або клас всередині збірки додатку. Розглянемо детальніше кожен з них:

- HomeController – відповідний за обробку запитів користувача, пов'язаних зі списком медалей;
- ProductController – відповідний за обробку запитів користувача, пов'язаних з конкретною медаллю;
- AccountController – відповідний за обробку запитів користувача, пов'язаних з реєстрацією та авторизацією;
- DialogueController – відповідний за обробку запитів користувача, пов'язаних з комунікацією між користувачами;
- Database – набір класів для роботи з базою даних;
- View – відповідає за формування html сторінок для користувача;
- Migrations – відповідає за оновлення бази даних.

Процеси авторизації, реєстрації та ведення каталогу користувачем наведені у діаграмі послідовності у частині графічного матеріалу.

					ДП ІС-6210.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

4.4 Специфікація функцій

Таблиця 4.1 – Опис публічних функцій програмного забезпечення

Назва	Опис
Клас AccountController	
[HttpGet] public IActionResult Login()	Повертає представлення для авторизації користувача.
[HttpPost] public async Task<IActionResult> Login(LoginModel model)	У якості параметра приймає об'єкт класу, який містить інформацію про логін та пароль користувача. Якщо логін та пароль вірні, повертає представлення головного каталогу та автентифікує користувача. Якщо логін та пароль не вірні, повертає представлення авторизації з повідомленням про помилку.
[HttpGet] public IActionResult Register()	Повертає представлення для реєстрації користувача.
[HttpPost] public async Task<IActionResult> Register(RegisterModel model)	У якості параметра приймає об'єкт класу, який містить інформацію про реєстраційні дані користувача. Якщо користувача з введеним логіном не існує, повертає представлення головного каталогу, реєструє та автентифікує користувача. Якщо користувач з введеним логіном існує, повертає представлення авторизації з повідомленням про помилку.
[HttpGet] public ViewResult MyAccount()	Повертає представлення особистого кабінету з інформацією про користувача.
Клас DialoguesController	

Продовження таблиці 4.1

Назва	Опис
[Authorize] [HttpGet] public ViewResult DialogueWith(int userId, int page=1)	У якості параметрів приймає унікальний ідентифікаційний номер користувача та номер сторінки. Повертає представлення діалогу з усіма повідомленнями.
[HttpPost] public IActionResult DialogueWith(string Text, int user2Id, int dialogueId)	У якості параметрів приймає текст повідомлення, ідентифікаційний номер користувача-отримувача та ідентифікаційний номер діалогу. Зберігає повідомлення до бази даних та повертає представлення діалогу вже з новим повідомленням.
[Authorize] public ViewResult MyDialogues()	Повертає представлення списку діалогів з усіма діалогами між користувачем та його співрозмовниками.
Клас HomeController	
[Authorize] public ViewResult MyCollection(int page = 1, string name = null , string category = null , string country = null , int? minYear = null , int? maxYear = null , string material = null , float? minWeight = null , float? maxWeight = null , float? minDiameter = null , float? maxDiameter = null , string byName = null , string byYear = null , string byCountry = null , string byCategory = null , string byMaterial = null , string byWeight = null , string byDiameter = null , string byDateTime= null)	У якості параметрів приймає інформацію про медалі. Фільтрує та сортує список медалей з власної колекції користувача за вхідною інформацією та повертає представлення колекції користувача зі списком медалей.

Продовження таблиці 4.1

Назва	Опис
<code>public ViewResult MainCatalog(int page = 1, string name = null, string category = null, string country = null, int? minYear = null, int? maxYear = null, string material = null, float? minWeight = null, float? maxWeight = null, float? minDiameter = null, float? maxDiameter = null, string byName = null, string byYear = null, string byCountry = null, string byCategory = null, string byMaterial = null, string byWeight = null, string byDiameter = null, string byDateTime = null)</code>	У якості параметрів приймає інформацію про медалі. Фільтрує та сортує список медалей головного каталогу за вхідною інформацією та повертає представлення головного каталогу зі списком медалей.
<code>public ViewResult UserCatalog(int page=1, string name=null, string category=null, string country = null, int? minYear = null, int? maxYear = null, string material = null, float? minWeight = null, float? maxWeight = null, float? minDiameter = null, float? maxDiameter = null, string byName = null, string byYear = null, string byCountry = null, string byCategory=null, string byMaterial = null, string byWeight = null, string byDiameter = null, string byDateTime = null)</code>	У якості параметрів приймає інформацію про медалі. Фільтрує та сортує список медалей користувацького каталогу за вхідною інформацією та повертає представлення користувацького каталогу зі списком медалей.

Продовження таблиці 4.1

Назва	Опис
<code>public ViewResult AboutUser (int userId, int page = 1, string name = null, string category = null, string country = null, int? minYear = null, int? maxYear = null, string material = null, float? minWeight = null, float? maxWeight = null, float? minDiameter = null, float? maxDiameter = null, string byName = null, string byYear = null, string byCountry = null, string byCategory = null, string byMaterial = null, string byWeight = null, string byDiameter = null, string byDateTime = null)</code>	У якості параметрів приймає інформацію про медалі та ідентифікаційний номер користувача. Фільтрує та сортує список медалей іншого користувача за вхідною інформацією та повертає представлення про іншого користувача зі списком медалей та інформацією про нього.
<code>public ViewResult UserSearch(string userName)</code>	У якості параметрів приймає запит з логіном користувача, якого шукають. Повертає представлення з результатами пошуку за запитом.
Клас ProductController	
<code>[HttpGet]</code> <code>[Authorize]</code> <code>public ViewResult AddProduct()</code>	Повертає представлення для додання нової медалі до застосунку.
<code>[HttpPost]</code> <code>[Authorize]</code> <code>public IActionResult AddProduct(AddProductModel product)</code>	У якості параметра приймає об'єкт класу, який містить інформацію про нову медаль. Якщо усі дані було введено вірно, то додає інформацію до бази даних (каталогу власної колекції користувача) та повертає представлення власної колекції користувача. Якщо дані введено невірно, повертає представлення додання нової медалі з повідомленнями про помилки.
<code>[HttpGet]</code> <code>[Authorize]</code> <code>public ViewResult UpdateProduct(int productId)</code>	У якості параметра приймає ідентифікаційний номер медалі. Повертає представлення для зміни інформації про медаль за номером.

Продовження таблиці 4.1

Назва	Опис
[HttpPost] public IActionResult UpdateProduct(AddProductModel product)	У якості параметра приймає об'єкт класу, який містить оновлену інформацію про медаль. Якщо усі дані було введено вірно, то зберігає інформацію до бази даних (каталогу власної колекції користувача) та повертає представлення власної колекції користувача. Якщо дані введено невірно, повертає представлення зміни інформації про медаль з повідомленнями про помилки.
[Authorize] public ViewResult DeleteProduct(int productId)	У якості параметра приймає ідентифікаційний номер медалі. Повертає представлення для видалення медалі з бази даних з інформацією про медаль.
public IActionResult DeleteProductAction(int productId)	У якості параметра приймає ідентифікаційний номер медалі. Видаляє медаль з бази даних та повертає представлення власної колекції користувача.
public ViewResult ProductInfo(int productId)	У якості параметра приймає ідентифікаційний номер медалі. Повертає представлення з інформацією про медаль.
[Authorize] public IActionResult AddToMyCollection(int productId)	У якості параметра приймає ідентифікаційний номер медалі. Додає медаль до каталогу власної колекції користувача. Повертає представлення власної колекції користувача з новою медаллю.
[Authorize] public IActionResult DeleteFromMyCollection(int productId)	У якості параметра приймає ідентифікаційний номер медалі. Видаляє медаль з каталогу власної колекції користувача. Повертає представлення власної колекції користувача з оновленим каталогом.

Продовження таблиці 4.1

Назва	Опис
[Authorize] public IActionResult AddToMainCatalog(int productId)	Функція доступна користувачам зі статусом «Адміністратор». У якості параметра приймає ідентифікаційний номер медалі. Додає медаль головного каталогу. Повертає представлення головного каталогу з новою медаллю.
[Authorize] public IActionResult DeleteFromMainCatalog(int productId)	Функція доступна користувачам зі статусом «Адміністратор». У якості параметра приймає ідентифікаційний номер медалі. Видаляє медаль з головного каталогу. Повертає представлення головного каталогу з оновленим каталогом.
[HttpGet] [Authorize] public ViewResult AddProductMC()	Функція доступна користувачам зі статусом «Адміністратор». Повертає представлення для додання нової медалі до головного каталогу.
[HttpPost] [Authorize] public IActionResult AddProductMC(AddProductModel product)	Функція доступна користувачам зі статусом «Адміністратор». У якості параметра приймає об'єкт класу, який містить інформацію про нову медаль. Якщо усі дані було введено вірно, то додає інформацію до бази даних (головного каталогу) та повертає представлення головного каталогу. Якщо дані введено невірно, повертає представлення додання нової медалі з повідомленнями про помилки.

Продовження таблиці 4.1

Назва	Опис
[HttpGet] [Authorize] public ViewResult UpdateProductMC(int productId)	Функція доступна користувачам зі статусом «Адміністратор». У якості параметра приймає ідентифікаційний номер медалі. Повертає представлення для зміни інформації про медаль з головного каталогу за номером.
[HttpPost] [Authorize] public IActionResult UpdateProductMC(AddProductModel product)	Функція доступна користувачам зі статусом «Адміністратор». У якості параметра приймає об'єкт класу, який містить оновлену інформацію про медаль. Якщо усі дані було введено вірно, то зберігає інформацію до бази даних (головного каталогу) та повертає представлення головного каталогу. Якщо дані введено невірно, повертає представлення зміни інформації про медаль з повідомленнями про помилки.

Висновок до розділу

У розділі програмного та технічного забезпечення були наведені засоби розробки застосунку, а саме: мову програмування C#, технологію ASP NET .Core MVC, фреймворк Entity Framework, мову тегів HTML, мову стилю сторінок CSS, та СУБД Microsoft SQL Server для збереження даних.

Також були висунуті вимоги до технічного забезпечення. Була описана архітектура програмного забезпечення, а також наведені діаграми для полегшеного розуміння процесу роботи застосування.

У розділі описані специфікації публічних функцій для опису їх роботи та призначення у тому чи іншому класі.

5. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

5.1.1 Опис головного каталогу

Для початку роботи з веб-застосунком користувач повинен запустити браузер та у рядку пошуку ввести <https://localhost:44301>. На головній сторінці застосунку користувач зможе побачити головний каталог сайту, представлений на рисунку 5.1

The screenshot shows the main catalog page with the following elements:

- Navigation Bar (Top):**
 - 1: Моя колекція
 - 2: Діалоги
 - 3: Головний каталог
 - 4: Користувацький каталог
 - 8: (Separator)
 - 7: (Separator)
 - 6: Авторизація
 - 5: (Separator)
- Main Content Area:**
 - Головний каталог: 7 медалей**
 - Результатів пошуку: 7
 - Table of Medals:**

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)	Користувач
9	Уранова промисловість 35 років	Чеська Республіка	Промисловість	1981	Бронза	60	50	MainAdmin
10	Ейнштейн 50 років зі смерті	Чеська Республіка	Наука	2005	Серебро	50	50	MainAdmin
	АЕС Кіншас	Китай	Промисловість	1998	Меднонікель	30	38	MainAdmin
	АЕС Лейбштадт	Швейцарія	Промисловість	1984	Бронза	140	60	MainAdmin
	Ейнштейн 5 франків	Швейцарія	Наука	1979	Меднонікель	30	30	MainAdmin
- Search and Filter Panel (Right):**
 - Пошук користувача: Знайти
 - Пошук медалі: Знайти
 - Фільтрація:**
 - Країна:
 - Категорія:
 - Рік: Від: До:
 - Матеріал:
 - Вага: Від: До:
 - Діаметр: Від: До:
 - Сортування:**
 - За датою додавання:
 - Не сортувати
 - За назвою:
 - Не сортувати

Рисунок 5.1 – Сторінка головного каталогу

Робота з застосунком починається саме зі сторінки головного каталогу. Вона показує список медалей, які ведуться адміністрацією сайту. Тут інформація повинна відповідати на 100% реальним екземплярам. Сторінка головного каталогу містить такі кнопки:

- «Моя колекція» (1): Посилання на сторінку з каталогом власної колекції користувача. Доступна тільки авторизованим користувачам;
- «Діалоги» (2): Посилання на список діалогів користувача. Доступна тільки авторизованим користувачам;
- «Головний каталог» (3): Посилання на сторінку з головним каталогом сайту;

- «Користувацький каталог» (4): Посилання на сторінку з каталогом медалей усіх користувачів сайту;
- «Авторизації» (5): Посилання на сторінку авторизації користувача. Замінює текст на логін користувача, коли користувач авторизований.

Якщо неавторизований користувач спробує використати кнопки «Моя колекція» та «Діалоги», застосунок відкриє сторінку з авторизацією.

Сторінка головного каталогу має наступні області:

Пошук користувача:

Пошук медалі:

Рисунок 5.2 – Пошук медалі або користувача

- пошук медалі або користувача (6). Дозволяє знайти медаль за назвою або користувача за логіном при заповненні відповідних форм та натисканням відповідної кнопки «Знайти»;

Фільтрація:

Країна

Категорія

Рік: Від: До:

Матеріал

Вага: Від: До:

Діаметр: Від: До:

Рисунок 5.3 – Фільтрація списку

- фільтрація списку медалей (7). Область заповнення даних для фільтрації списку медалей. Для застосування фільтрації потрібно натиснути кнопку «Прийняти» під областю сортування;

Сортування:

За датою додавання:

Не сортувати

За назвою:

Не сортувати

За країною:

Не сортувати

За категорією:

Не сортувати

За роком:

Не сортувати

За матеріалом:

Не сортувати

За вагою:

Не сортувати

За діаметром:

Не сортувати

Прийняти

Рисунок 5.4 – Сортування списку

- сортування списку за параметрами (8). Область для заповнення даних для сортування списку медалей за параметрами. Для застосування сортування потрібно натиснути кнопку «Прийняти». Сортування може бути виконано за декількома параметрами.

Сторінка головного каталогу містить таблицю зі списком медалей, що ведеться адміністрацією. Таблиця показує наступні параметри медалей: фото, назва, країна випуску, категорія, рік випуску, матеріал, вага, діаметр та логін користувача, що додав медаль до сайту. Назва містить у собі посилання на сторінку медалі (9), де можна побачити збільшене фото та більш детальний

опис. Логін користувача, що додав медаль до сайту, містить у собі посилання на сторінку користувача (10).

На одній сторінці головного каталогу поміщається до ста медалей. Якщо медалей більше ніж сто, створюється нова сторінка та посилання на неї.

5.1.2 Опис авторизації

Для використання усіх можливостей системи, користувачу необхідно авторизуватися. На рисунку 5.5 показана сторінка авторизації:

Рис. 5.5. Сторінка авторизації. Елементи позначені цифрами: 1 - поле для введення Email, 2 - кнопка «Увійти», 3 - посилання «Реєстрація».

Рисунок 5.5 – Сторінка авторизації

Сторінка авторизації містить:

- область для введення логіну та паролю (1);
- кнопку «Увійти» (2). При натисканні, якщо логін та пароль введено вірно, система авторизує користувача та повертає на сторінку головного каталогу. Якщо дані введено невірно, система повідомляє про це користувачу;
- посилання на сторінку реєстрації (3).

5.1.3 Опис реєстрації

Для можливості авторизації у системі користувачу, потрібно зареєструватися. На рисунку 5.6 показана сторінка реєстрації:

The image shows a web form titled "Регістрація" (Registration). It contains several input fields: "Email:", "Ім'я:" (Name), "Прізвище:" (Surname), "Логін (для відображення на сайті)" (Login), "Пароль:" (Password), and "Повторіть пароль:" (Repeat password). Below these fields is a button labeled "Регістрація". A large number "1" with a line pointing to the form fields is located above the form. A large number "2" with a line pointing to the registration button is located to the left of the button.

Рисунок 5.6 – Сторінка реєстрації

Сторінка реєстрації містить:

- область для введення реєстраційних даних (1);
- кнопка «Регістрація» (2). При натисканні, якщо усі поля заповнені, система реєструє та авторизує користувача, та повертає на сторінку головного каталогу. Якщо є пусті поля, система повідомляє про це користувачу.

5.1.4 Опис власної колекції

Авторизований користувач має можливість вести каталог власної колекції. Для цього йому потрібно перейти на сторінку «Моя колекція». З самого початку список медалей буде пустим. На рисунку 5.7 показана сторінка каталогу власної колекції з заповненим списком медалей:










Моя колекція Діалоги Головний каталог Користувачський каталог

kravez95@gmail.com

Моя колекція: 3 медалей

Результатів пошуку: 3

[Додати медаль](#)

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)		
	Вільгельм Рентген 150 років	Німеччина	Наука	1996	Серебро	22	35		
	Вільгельм Рентген	Німеччина	Наука	1978	Серебро	31	39		
	Хіросіма	Німеччина	Історія та особистості	2015	Серебро	26	38		

Пошук користувача: [Знайти](#)

Пошук медалі: [Знайти](#)

Фільтрація:

Країна:

Категорія:

Рік: Від: До:

Матеріал:

Вага: Від: До:

Діаметр: Від: До:

Сортування:

За датою додавання:

Не сортувати

Рисунок 5.7 – Сторінка каталогу власної колекції користувача

Сторінка надає можливість користувачу ведення каталогу власної колекції за допомогою наступних кнопок:

- «Додати медаль» (1). При натисканні відкривається сторінка для додання медалі;
- «Змінити інформацію про медаль» (2). Доступна для кожної медалі з каталогу. При натисканні відкривається сторінка для зміни інформації про медаль;
- «Видалити медаль» (3). Доступна для кожної медалі з каталогу. При натисканні відкривається сторінка з підтвердженням про видалення медалі.

Для більш детального опису сторінки каталогу власної колекції дивитись «Опис головного каталогу».

5.1.5 Опис додання медалі

Для поповнення власної колекції користувачеві необхідно додавати інформацію про медаль, яка у нього є. Для цього необхідна сторінка додання медалі, яка представлена на рисунку 5.8:

Скріншот сторінки «Додання медалі» з елементами форматування та нумерацією:

- 1** – область заповнення інформації про медаль (форма з полями: Назва, Країна, Категорія, Рік, Матеріал, Вага (гр), Діаметр (мм), Опис).
- 2** – кнопка «Завантаження фото» (іконка стрілки вгору).
- 3** – кнопка «Додати».

Рисунок 5.8 – Сторінка додання медалі

Сторінка додання медалі містить:

- область заповнення інформації про медаль (1);
- кнопку «Завантаження фото» (2);
- кнопку «Додати». При натисканні, якщо усі поля заповнені та

розмір фото менше ніж 5 мегабайт, система збереже інформацію до бази даних та поверне користувача на сторінку власної колекції. Якщо є пусте поле, система повідомить про це користувачу. Якщо розмір фото більше ніж 5 мегабайт, система повідомить про це користувачу.

5.1.6 Опис зміни інформації про медаль

Якщо є потреба зміни інформації про медаль чи оновлення фотографії, користувач може це зробити за допомогою сторінки зміни інформації про медаль, яка представлена на рисунку 5.9:

Зміна:

Назва:

Німеччина ▾

Наука ▾

Рік:

Серебро ▾

Вага (гр):

Діаметр (мм):

Опис:

Вибрати файл

Зберегти







Сторінка зміни інформації дуже схожа на сторінку додання медалі (див «Опис додання медалі»). Головною різницею є те, що користувачу доступна інформація про медаль, яку він змінює, у полях введення. Це запобігає проблемі повторного введення інформації.

5.1.7 Опис користувацького каталогу

Для перегляду медалей, які є у інших користувачів існує користувацький каталог. Сторінка користувацького каталогу представлена на рисунку 5.10:

Моя колекція
Діалоги
Головний каталог
Користувацький каталог
kravez95@gmail.com

Користувацький каталог: 9 медалей
Результатів пошуку: 9

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)	Користувач
	Роберт Опенгеймер	Франція	Наука	1988	Бронза	130	60	mykola.ryasniy
	АЕС Шуз	Франція	Промисловість	1991	Бронза	120	60	mykola.ryasniy
	АЕС Блас	Франція	Промисловість	1998	Бронза	120	60	mykola.ryasniy
	Марія Кюрі. Великі французи	Франція	Наука	2015	Серебро	20	35	ivan.kazimir
	Інститут Кюрі 50 євро	Франція	Наука	2009	Серебро	60	50	ivan.kazimir
	Марія Кюрі	Франція	Наука	2000	Золото	1	14	ivan.kazimir

Пошук користувача:

Пошук медалі:

Фільтрація:

Країна:

Категорія:

Рік: Від: До:

Матеріал:

Вага: Від: До:

Діаметр: Від: До:

Сортування:

За датою додавання:

За назвою:

За країною:

За категорією:

https://localhost:44301/Home/MyCollection

Рисунок 5.10 – Сторінка користувацького каталогу

Сторінка користувацького каталогу схожа на сторінку головного каталогу(див. «Опис головного каталогу»). Тут інформація може бути недостовірною, так як напряду залежить від користувачів. Каталог показує усі медалі, які користувачі власноруч додавали до своїх колекцій.

5.1.8 Опис списку діалогів

Для комунікації між користувачами система підтримує діалоги між ними. Якщо виникає потреба дізнатися контактні дані у користувача або уточнити інформацію, є можливість написати йому. На рисунку 5.11 показана сторінка списку діалогів з іншими користувачами:

Моя колекція Діалоги Головний каталог Користувацький каталог

Мої діалоги з:


- 1 — mykola.gyasniy 
- 2 — ivan.kazimir

Рисунок 5.11 – Сторінка списку діалогів

Список діалогів складається зі списку логінів користувачів, з якими був початий діалог. Якщо в діалозі є непрочитані повідомлення, поряд з логіном користувача-співрозмовника буде зображення листа (1). Це корисно, коли тільки був створений діалог та користувач вперше написав. Кожен логін містить у собі посилання на діалог з цим користувачем (2).

5.1.9 Опис діалогу

На рисунку 5.12 зображена сторінка діалогу з іншим користувачем:

Моя колекція Діалоги Головний каталог Користувацький каталог

1 — Діалог з ivan.kazimir

2 — 14 мая 2020 г.
petro.kravez: Доброго дня! Про яку саме?

14 мая 2020 г.
ivan.kazimir: Привіт. Я хочу дещо дізнатися про медаль Опенгеймера

3 —

①

Рисунок 5.12 – Сторінка діалогу з іншим користувачем

Сторінка містить:

- посилання на сторінку користувача, з яким ведеться діалог (1);
- область повідомлень між користувачами (2). Для зручності повідомлення від різних користувачів виділені різним кольором;
- область для введення та відправлення повідомлення.

5.1.10 Опис інформації про медаль

У кожної медалі є власна сторінка, де можна побачити збільшене фото та більш детальну інформацію. На рисунку 5.13 зображена сторінка інформації про медаль «АЕС Кіншан»:

1



Назва: АЕС Кіншан

Країна: Китай

Рік: 1998

Матеріал: Меднонікель

Вага: 30

Діаметр: 38

2

Додати в мою колекцію

3

Ця медаль є у:

Користувачів не знайдено

Рисунок 5.13 – Сторінка медалі «АЕС Кіншан»

Сторінка медалі містить:

- область з інформацією про медаль (1);
- кнопку «Додати в мою колекцію»(2), якщо медаль належить головному каталогу. При натисканні медаль додається у каталог власної колекції користувача;
- список логінів користувачів, в кого є ця медаль. Логіни містять у собі посилання на сторінки користувачів.

Якщо медаль була додана до власної колекції, то на місті кнопки «Додати в мою колекцію» буде кнопка «Видалити з моєї колекції», яка дозволить видалити медаль з каталогу колекції користувача.

5.1.11 Опис інформації про користувача

Кожен користувач має свою сторінку з інформацією про нього та каталогом його колекцію, доступну усім іншим користувачам. На рисунку 5.14 зображено сторінку користувача з логіном mykola.ryasniy:

					ДП ІС-6210.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44




Моя колекція
Діалоги
Головний каталог
Користувацький каталог
kravez95@gmail.com

Логін: mykola.gyasniy
Статус: Користувач
Медалей в каталозі: 3
Результатів пошуку: 3

Пошук користувача:

Пошук медалі:

Фільтрація:
Країна
Категорія
Рік: Від: До:
Матеріал
Вага: Від: До:
Діаметр: Від: До:
Сортування:
За датою додавання:
За назвою:
За країною:

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)	Користувач
	Роберт Опенгеймер	Франція	Наука	1988	Бронза	130	60	mykola.gyasniy
	АЕС Шуз	Франція	Промисловість	1991	Бронза	120	60	mykola.gyasniy
	АЕС Блас	Франція	Промисловість	1998	Бронза	120	60	mykola.gyasniy

1

Рисунок 5.14 – Сторінка користувача

На сторінці можна побачити список медалей колекції цього користувача. Є можливість написати користувачеві за допомогою кнопки «Написати повідомлення».

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій системи підтримки діяльності фалеристів вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Тести включають перевірку основних функцій системи, таких як:

- авторизація\реєстрація користувача;
- навігація сайтом;
- ведення каталогу власної колекції;
- фільтрація та сортування каталогів;
- написання повідомлення іншому користувачу.

Результати проходження тестів наведені в таблицях 5.1-5.16:

Таблиця 5.1 – Тестування сторінки авторизації на невірні логін та пароль.

Тест:	Форма авторизації перевіряє чи існує користувач з введеним логіном та паролем.
Початковий стан системи:	Відкрита сторінка авторизації.
Дія:	Вводимо невірні логін та пароль. Натискаємо на кнопку «Увійти».
Очікуваний результат:	Під полями вводу логіну та паролю з'являється повідомлення «Некоректні логін і (або) пароль».
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.2 – Тестування сторінки авторизації на вірні логін та пароль.

Тест:	Форма авторизації перевіряє чи існує користувач з введеним логіном та паролем.
Початковий стан системи:	Відкрита сторінка авторизації.
Дія:	Вводимо вірні логін та пароль. Натискаємо на кнопку «Увійти».
Очікуваний результат:	Відкривається сторінка головного каталогу. Кнопка «Авторизація» замінюється на кнопку особистого кабінету.
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.3 – Тестування сторінки реєстрації на порожні поля вводу.

Тест:	Форма реєстрації перевіряє чи є незаповнені поля.
Початковий стан системи:	Відкрита сторінка реєстрації.
Дія:	Заповнюємо поля «Email», «Ім'я», «Логін», «Пароль», «Повторіть пароль». Залишаємо порожнім поле «Прізвище». Натискаємо кнопку «Реєстрація».
Очікуваний результат:	Біля поля «Прізвище» з'являється повідомлення «Не вказане прізвище».
Фактичний результат співпадає з очікуваним:	Так

Таблиця 5.4 – Тестування сторінки реєстрації на заповнені поля вводу.

Тест:	Форма реєстрації реєструє та авторизує користувача у системі.
Початковий стан системи:	Відкрита сторінка реєстрації.
Дія:	Заповнюємо поля «Email», «Ім'я», «Прізвище», «Логін», «Пароль», «Повторіть пароль». Натискаємо кнопку «Реєстрація».
Очікуваний результат:	Відкривається сторінка головного каталогу. Кнопка «Авторизація» замінюється на кнопку особистого кабінету.
Фактичний результат співпадає з очікуваним:	Так

Таблиця 5.5 – Тестування сторінки додання медалі до каталогу власної колекції з порожніми полями.

Тест:	Форма додання медалі перевіряє чи є порожні поля, що є обов'язковими.
Початковий стан системи:	Відкрита сторінка додання медалі.
Дія:	Заповнюємо поля «Назва», «Країна», «Категорія», «Рік», «Матеріал», «Діаметр». Залишаємо порожнім поле «Вага». Натискаємо кнопку «Додати».
Очікуваний результат:	Біля поля «Вага» з'являється повідомлення «Не вказана вага».
Фактичний результат співпадає з очікуванням:	Так

Таблиця 5.6 - Тестування сторінки додання медалі до каталогу власної колекції з заповненими полями.

Тест:	Форма додання медаль до бази даних.
Початковий стан системи:	Відкрита сторінка додання медалі.
Дія:	Заповнюємо поля «Назва», «Країна», «Категорія», «Рік», «Матеріал», «Діаметр», «Вага». Натискаємо кнопку «Додати».
Очікуваний результат:	Відкривається сторінка каталогу власної колекції. У списку медалей знаходиться нова медаль з заданою інформацією.
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.7 - Тестування сторінки зміни інформації про медаль з порожніми полями.

Тест:	Форма зміни інформації про медаль перевіряє чи є порожні поля, що є обов'язковими.
Початковий стан системи:	Відкрита сторінка зміни інформації про медаль. Усі поля вводу заповнені інформацією про відповідний параметр.
Дія:	Видаляємо текст з поля «Вага». Натискаємо кнопку «Зберегти».
Очікуваний результат:	Біля поля «Вага» з'являється повідомлення «Не вказана вага».
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.8 - Тестування сторінки зміни інформації про медаль з заповненими полями.

Тест:	Форма зміни інформації про медаль зберігає нову інформацію про медаль до бази даних.
Початковий стан системи:	Відкрита сторінка зміни інформації про медаль. Усі поля вводу заповнені інформацією про відповідний параметр.
Дія:	Змінюємо вміст поля «Вага». Натискаємо кнопку «Зберегти».
Очікуваний результат:	Відкривається сторінка каталогу власної колекції. Поле вага, змінюваної медалі, має новий вміст.
Фактичний результат співпадає з очікуванням:	Так.

Таблиця 5.9 – Тестування кнопки «Моя колекція» для неавторизованого користувача.

Тест:	Посилання на сторінку каталогу власної колекції перенаправляє неавторизованого користувача на сторінку авторизації.
Початковий стан системи:	Відкрита сторінка головного каталогу. Користувач неавторизований.
Дія:	Натискаємо на кнопку «Моя колекція».
Очікуваний результат:	Відкривається сторінка авторизації.
Фактичний результат співпадає з очікуванням:	Так.

Таблиця 5.10 – Тестування кнопки «Моя колекція» для авторизованого користувача.

Тест:	Посилання на сторінку каталогу власної колекції перенаправляє авторизованого користувача на сторінку каталогу власної колекції.
Початковий стан системи:	Відкрита сторінка головного каталогу. Користувач авторизований.
Дія:	Натискаємо на кнопку «Моя колекція».
Очікуваний результат:	Відкривається сторінка каталогу власної колекції користувача.
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.11 – Тестування кнопки «Діалоги» для неавторизованого користувача.

Тест:	Посилання на сторінку списку діалогів перенаправляє неавторизованого користувача на сторінку авторизації.
Початковий стан системи:	Відкрита сторінка головного каталогу. Користувач неавторизований.
Дія:	Натискаємо на кнопку «Діалоги».
Очікуваний результат:	Відкривається сторінка авторизації.
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.12 - Тестування кнопки «Діалоги» для авторизованого користувача.

Тест:	Посилання на сторінку списку діалогів перенаправляє авторизованого користувача на сторінку списку діалогів.
Початковий стан системи:	Відкрита сторінка головного каталогу. Користувач авторизований.
Дія:	Натискаємо на кнопку «Діалоги».
Очікуваний результат:	Відкривається сторінка списку діалогів користувача.
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.13 – Тестування переходу на сторінку медалі за посиланням.

Тест:	Назва медалі у списку головного каталогу перенаправляє користувача на сторінку цієї медалі.
Початковий стан системи:	Відкрита сторінка головного каталогу.
Дія:	Натискаємо на назву медалі «АЕС Кіншан».
Очікуваний результат:	Відкривається сторінка медалі «АЕС Кіншан» з інформацією про неї.
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.14 – Тестування переходу на сторінку іншого користувача.

Тест:	Логін користувача, що додав медаль, у списку головного каталогу перенаправляє на сторінку цього користувача.
Початковий стан системи:	Відкрита сторінка головного каталогу.
Дія:	Натискаємо на логін користувача «MainAdmin».
Очікуваний результат:	Відкривається сторінка з інформацією про користувача «MainAdmin».
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.15 – Тестування фільтрації головного каталогу.

Тест:	Сторінка головного каталогу показує відфільтрований список медалей.
Початковий стан системи:	Відкрита сторінка головного каталогу.
Дія:	Заповнюємо поле «Країна» вмістом «Швейцарія».
Очікуваний результат:	На сторінці головного каталогу будуть відображені медалі, які мають параметр «Країна» рівним «Швейцарія».
Фактичний результат співпадає з очікуваним:	Так.

Таблиця 5.16 – Тестування сортування головного каталогу.

Тест:	Сторінка головного каталогу показує відсортований список медалей.
Початковий стан системи:	Відкрита сторінка головного каталогу.
Дія:	Заповнюємо поле «Сортування за країною» вмістом «За зростанням».
Очікуваний результат:	На сторінці головного каталогу будуть відображені медалі, відсортовані за країною у порядку зростання.
Фактичний результат співпадає з очікуванням:	Так.

Висновок до розділу

У технологічному розділі було детально розглянуто керівництво користувача та опис проведених тестів.

Керівництво користувача має детальний опис початку роботи із системою, важливі деталі щодо користування власним каталогом та наведені пункти щодо необхідності реєстрації та авторизації. Були приведені скріншоти роботи застосунку для пояснення використання різних форм та сторінок. На скріншотах наведені символічні позначення для тих чи інших елементів сторінки.

Також у розділі були наведені тести, які були виконані, для перевірки працездатності системи. Перш за все, було протестовано можливість реєстрації та авторизації користувача. Також важливою частиною роботи є правильна навігація по сайту. Також тести охопили роботу фільтрації та сортування списку медалей у каталогах.

ЗАГАЛЬНІ ВИСНОВКИ

У пояснювальній записці наведено детальний опис дипломного проекту, який присвячено системі підтримки діяльності фалеристів.

У розділі загальних положень було розглянуто цілі та головні задачі розробки системи, опис предметного середовища, огляд аналогів існуючих рішень та опис функціональної моделі. Було наведено інформацію про особливості колекціонування настільних медалей та їх параметри. Дана робота присвячена підтримки діяльності колекціонерів настільних медалей, а саме спрощенню процесу експонування за рахунок розміщення та зберігання колекції на сайті. Було розглянуто аналоги існуючих рішень, виявлено їх недоліки та переваги. З урахуванням недоліків було зроблено висновок про необхідність та актуальність розробки системи. У розділі було надано опис дійових осіб та їх головні призначення. За допомогою структурної схеми використань можна побачити функції, які будуть виконуватися дійовими особами. Був проведений аналіз потреб колекціонерів для формування конкретних задач, які необхідно вирішити для розробки застосування.

У розділі інформаційного забезпечення було описано вхідні дані для виконання задачі підтримки діяльності фалеристів, а саме: дані для наповненості сайту, дані для першого користувача зі статусом «Адміністратор» та дані користувача, діяльність якого буде підтримувати система. Вихідними даними є результати запитів користувача за пошуком, фільтрацією та сортуванням, та контент каталогів сайту. У розділі було описано структуру бази даних та призначення кожної таблиці. Також було наведено опис кожної таблиці та опис кожного стовбця.

В розділі математичного забезпечення наведено змістовну та математичну постановку задачі пошуку назви наближеної до запиту користувача. Було наведено три алгоритми для вирішення поставленої задачі. За результатами аналізу був вибраний метод пошуку найбільшого підрядка

					ДП ІС-6210.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

двох рядків. Наведено детальний опис алгоритму та структурну схему його роботи для розв'язання задачі.

У розділі програмного та технічного забезпечення були наведені засоби розробки застосунку, а саме: мову програмування C#, технологію ASP NET .Core MVC, фреймворк Entity Framework, мову тегів HTML, мову стилю сторінок CSS, та СУБД Microsoft SQL Server для збереження даних. Також були висунуті вимоги до технічного забезпечення. Була описана архітектура програмного забезпечення, а також наведені діаграми для полегшеного розуміння процесу роботи застосування. У розділі описані специфікації публічних функцій для опису їх роботи та призначення у тому чи іншому класі.

У технологічному розділі було детально розглянуто керівництво користувача та опис проведених тестів. Керівництво користувача має детальний опис початку роботи із системою, важливі деталі щодо користування власним каталогом та наведені пункти щодо необхідності реєстрації та авторизації. Були приведені скріншоти роботи застосунку для пояснення використання різних форм та сторінок. На скріншотах наведені символічні позначення для тих чи інших елементів сторінки. Також у розділі були наведені тести, які були виконані, для перевірки працездатності системи. Перш за все, було протестовано можливість реєстрації та авторизації користувача. Також важливою частиною роботи є правильна навігація по сайту. Також тести охопили роботу фільтрації та сортування списку медалей у каталогах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Райдо знак. Настільні медалі [Електронний ресурс]. 2020 - Режим доступу до ресурсу: <http://raido-znak.com.ua/ua/medali-/nastolnie-medali->
2. Expert Coders. Поиск длины наибольшей общей подпоследовательности [Електронний ресурс]. 2020 - Режим доступу до ресурсу: <http://coders.ask-ru.net/алгоритм-поиска-длины-наибольшей-общ/>
3. GeeksforGeeks. Longest Common Substring [Електронний ресурс]. 2020 - Режим доступу до ресурсу: <https://www.geeksforgeeks.org/longest-common-substring-dp-29/#:~:text=Dynamic%20Programming%20can%20be%20used,these%20lengths%20in%20a%20table.>
4. Герберт Шилдт, С# 4.0: полное руководство.: Пер. з англ. – Москва.: ТОВ «І.Д. Вільямс», 2011. – 1056 с.
5. Адам Фрімен, ASP.NET Core MVC 2 с примерами на С# для профессионалов. 7-е видання.: Пер. з англ. – Санкт-Петербург.: ТОВ «Діалектика», 2019. – 1008 с.
6. Крістіан Нейгел, Білл Ів'єн, Джей Глінн, Карлі Уотсон, Морган Скіннер, С# 5.0 и платформа .NET 4.5 для профессионалов.: Пер. з англ. – Москва. ТОВ «І.Д. Вільямс», 2014. – 1440 с.
7. Роберт Вийера, Программирование баз данных Microsoft SQL Server 2005 для профессионалов.: Пер. з англ. – Москва.: ТОВ «І.Д. Вільямс», 2008. – 1072 с.
8. Український веб-довідник [Електронний ресурс]. 2020 - Режим доступу до ресурсу: <https://css.in.ua>
9. Бен Фрейн, HTML5 та CSS3. Разработка сайтов для любых браузеров и устройств. – Санкт-Петербург.: ТОВ «Пітер», 2014. – 304 с.

ДОДАТОК А

*Тексти програмного коду**Інформаційна система підтримки діяльності фалеристів*

(найменування програми (документа))

DVD-R

(Вид носія даних)

30 арк. 43 Кб

(Обсяг програми (документа), арк.,) Кб)

Київ - 2020 року

					ДП ІС-6210.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

AccountController.cs

```

using MedalsWebSystem.Models;
using MedalsWebSystem.ViewModels;
using Microsoft.AspNetCore.Authentication;
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;

namespace MedalsWebSystem.Controllers
{
    public class AccountController : Controller
    {
        private ApplicationDbContext db;
        public AccountController(ApplicationContext context)
        {
            db = context;
        }
        [HttpGet]
        public IActionResult Login()
        {
            ViewBag.UserAuth = User.Identity.IsAuthenticated;
            ViewBag.UserName = User.Identity.Name;
            return View();
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> Login(LoginModel model)
        {
            if (ModelState.IsValid)
            {
                User user = await db.Users.FirstOrDefaultAsync(u => u.Email ==
model.Email && u.Pass == model.Password);
                if (user != null)
                {
                    await Authenticate(user);

                    return RedirectToAction("MainCatalog", "Home");
                }
                ModelState.AddModelError("", "Некоректні логін і (або) пароль");
            }
            return View(model);
        }
        [HttpGet]
        public IActionResult Register()
        {
            return View();
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> Register(RegisterModel model)
        {
            if (ModelState.IsValid)
            {
                User user = await db.Users.FirstOrDefaultAsync(u => u.Email ==
model.Email);

```

```
        if (user == null)
        {
            user = new User { Email = model.Email, Pass = model.Password,
Name=model.Name, Surname=model.Surname, Login=model.Login, Role="Користувач" };
            db.Users.Add(user);
            await db.SaveChangesAsync();

            await Authenticate(user);

            return RedirectToAction("MainCatalog", "Home");
        }
        else
        {
            ModelState.AddModelError("", "Некоректні логін і (або) пароль");
        }
        return View(model);
    }

    [HttpGet]
    public IActionResult MyAccount()
    {
        ViewBag.UserAuth = User.Identity.IsAuthenticated;
        ViewBag.UserName = User.Identity.Name;
        User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);
        return View(user);
    }

    private async Task Authenticate(User user)
    {
        var claims = new List<Claim>
        {
            new Claim(ClaimsIdentity.DefaultNameClaimType, user.Email),
        };

        ClaimsIdentity id = new ClaimsIdentity(claims, "ApplicationCookie",
ClaimsIdentity.DefaultNameClaimType, ClaimsIdentity.DefaultRoleClaimType);

        await
HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, new
ClaimsPrincipal(id));
    }

    public async Task<IActionResult> Logout()
    {
        await
HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
        return RedirectToAction("MainCatalog", "Home");
    }
}
```


DialoguesController.cs

```

using MedalsWebSystem.Models;
using MedalsWebSystem.ViewModels;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Xml.Linq;

namespace MedalsWebSystem.Controllers
{
    public class DialoguesController : Controller
    {
        private ApplicationDbContext db;
        public DialoguesController(ApplicationContext context)
        {
            db = context;
        }

        [Authorize]
        [HttpGet]
        public IActionResult DialogueWith(int userId, int page=1)
        {
            int pageSize = 100;
            ViewBag.UserAuth = User.Identity.IsAuthenticated;
            ViewBag.UserName = User.Identity.Name;

            User user1 = db.Users.Include(xc => xc.UserDialogues).ThenInclude(xc =>
xc.Dialogue).FirstOrDefault(x => x.Email == User.Identity.Name);
            User user2 = db.Users.Include(xc => xc.UserDialogues).ThenInclude(xc =>
xc.Dialogue).FirstOrDefault(x => x.UserId == userId);
            int dialogueId = FindDialogueId(user1, user2);

            Dialogue dialogue = db.Dialogues.Include(xc => xc.Messages).ThenInclude(x =>
x.Sender).FirstOrDefault(x => x.DialogueId == dialogueId);
            Dialogue dialogue2 = db.Dialogues.Include(xc =>
xc.UserDialogues).FirstOrDefault(x => x.DialogueId == dialogueId);
            List<Message> messages = dialogue.Messages.OrderByDescending(x =>
x.Time).ToList();
            UserDialogue ud = dialogue2.UserDialogues.FirstOrDefault(x => x.UserId ==
user1.UserId);
            ud.IsRead = true;
            db.Update(ud);
            db.SaveChanges();

            var count = messages.Count();
            ViewBag.SearchCount = count;
            var items = messages.Skip((page - 1) * pageSize).Take(pageSize).ToList();
            PageViewModel pageViewModel = new PageViewModel(count, page, pageSize);

            DialogueViewModel d1 = new DialogueViewModel
            {
                Dialogue = dialogue,
                User2 = user2,
                Messages = messages,
                PageViewModel = pageViewModel
            };
        }
    }
}

```

```

        return View(d1);
    }

    public IActionResult DialogueWith(string Text, int user2Id, int dialogueId)
    {
        if(Text.Length>250)
        {
            Text = Text.Substring(250);
        }
        User user1 = db.Users.Include(xc => xc.UserDialogues).ThenInclude(xc =>
xc.Dialogue).FirstOrDefault(x => x.Email == User.Identity.Name);
        Message message = new Message {Text=Text, Time= DateTime.UtcNow , Sender=
user1 };
        Dialogue dialogue =
db.Dialogues.Include(xc=>xc.UserDialogues).FirstOrDefault(x => x.DialogueId ==
dialogueId);
        UserDialogue ud = dialogue.UserDialogues.FirstOrDefault(x => x.UserId ==
user2Id);
        ud.IsRead = false;
        db.Update(ud);
        dialogue.Messages = new List<Message>();
        dialogue.Messages.Add(message);
        db.Update(dialogue);
        db.SaveChanges();
        return RedirectToAction("DialogueWith", new { userId = user2Id });
    }

    private int FindDialogueId(User user1, User user2)
    {
        List<Dialogue> dialogues1 = user1.UserDialogues.Select(x =>
x.Dialogue).ToList();
        List<Dialogue> dialogues2 = user2.UserDialogues.Select(x =>
x.Dialogue).ToList();

        foreach(Dialogue d1 in dialogues1)
        {
            foreach(Dialogue d2 in dialogues2)
            {
                if (d1.DialogueId == d2.DialogueId)
                    return d1.DialogueId;
            }
        }

        Dialogue newD = new Dialogue();
        db.Dialogues.Add(newD);
        db.SaveChanges();

        user1.UserDialogues.Add(new UserDialogue { UserId = user1.UserId, DialogueId
= newD.DialogueId });
        user2.UserDialogues.Add(new UserDialogue { UserId = user2.UserId, DialogueId
= newD.DialogueId });
        db.SaveChanges();
        return newD.DialogueId;
    }

    [Authorize]
    public ViewResult MyDialogues()
    {
        ViewBag.UserAuth = User.Identity.IsAuthenticated;
        ViewBag.UserName = User.Identity.Name;
    }

```

```
User user1 =
db.Users.Include(xc=>xc.UserDialogues).ThenInclude(xc=>xc.Dialogue).FirstOrDefault(x =>
x.Email == User.Identity.Name);

List <Dialogue> mainDialogues = new List<Dialogue>();
List<UserAndRead> usersAndRead = new List <UserAndRead>();
var dialogues = user1.UserDialogues.Select(xc => xc.Dialogue).ToList();
foreach(var d in dialogues)
{

mainDialogues.Add(db.Dialogues.Include(xc=>xc.UserDialogues).ThenInclude(xc=>xc.User).Fir
stOrDefault(x => x.DialogueId == d.DialogueId));
}
foreach(Dialogue d in mainDialogues)
{
    UserDialogue ud = d.UserDialogues.FirstOrDefault(x => x.UserId ==
user1.UserId);
    bool IsRead = ud.IsRead;

    ud = d.UserDialogues.FirstOrDefault(x => x.UserId != user1.UserId);
    User user = ud.User;
    usersAndRead.Add(new UserAndRead { User = user, IsRead = IsRead });
}
usersAndRead = usersAndRead.OrderBy(x => x.IsRead).ToList();
return View(usersAndRead);
}
}
```

HomeController.cs

```

using MedalsWebSystem.Models;
using MedalsWebSystem.ViewModels;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Security.Cryptography.X509Certificates;
using System.Threading.Tasks;

namespace MedalsWebSystem.Controllers
{
    public class HomeController : Controller
    {
        private ApplicationContext db;
        private FilterInformation filterInformation;
        private SortingInformation sortingInformation;

        public string[] RusCountries { get; } = new string[] { "Китай", "Андорра",
"Аргентина", "Австрія", "Білорусь", "Бельгія", "Болгарія",
"Канада", "Хорватія", "Куба", "Кіпр", "Чеська Республіка", "Данія", "Еквадор",
"Естонія", "Фінляндія", "Франція",
"Німеччина", "Греція", "Гватемала", "Угорщина", "Ірландія", "Острів Мен",
"Італія", "Японія", "Казахстан", "Латвія",
"Литва", "Люксембург", "Македонія", "Мальта", "Мексика", "Молдавія", "Монако",
"Нідерланди", "Нікарагуа", "Норвегія",
"Парагвай", "Перу", "Польща", "Португалія", "Румунія", "Росія", "Сан Марино",
"Сербія", "Словаччина", "Словенія",
"Південна Корея", "Іспанія", "Швеція", "Швейцарія", "Туреччина", "Україна",
"Великобританія", "США", "Ватикан"};
        public string[] RusMaterials { get; } = new string[] { "Золото", "Серебро",
"Мідь", "Бронза", "Бронза", "Томпак", "Алюміній",
"Меднонікель", "Нейзильбер", "Сталь", "Біметал", "Інше"};
        public string[] RusCategories { get; } = new string[] { "Архітектура",
"Мистецтво", "Спорт", "Наука", "Промисловість",
"Транспорт", "Фауна", "Флора", "Освоєння космосу", "Історія та особистості",
"Інше"};

        public HomeController(ApplicationContext context)
        {
            db = context;
        }
        [Authorize]
        public IActionResult MyCollection(int page = 1, string name = null, string category
= null, string country = null, int? minYear = null, int? maxYear = null, string material
= null,
            float? minWeight = null, float? maxWeight = null, float? minDiameter = null,
float? maxDiameter = null,
            string byName = null, string byYear = null, string byCountry = null, string
byCategory = null, string byMaterial = null, string byWeight = null, string byDiameter =
null,
            string byDateTime=null)
        {
            ViewBag.UserAuth = User.Identity.IsAuthenticated;
            ViewBag.UserName = User.Identity.Name;
            Array.Sort(RusCountries);
            ViewBag.Countries = RusCountries;
            ViewBag.Materials = RusMaterials;
            ViewBag.Categories = RusCategories;

```

```

        User user =
db.Users.Include(c=>c.UserProducts).ThenInclude(cs=>cs.Product).FirstOrDefault(x =>
x.Email == User.Identity.Name);
        List<Product> allProducts = user.UserProducts.Select(sc =>
sc.Product).OrderByDescending(x => x.DateTime).ToList();
        ViewBag.CatalogCount = allProducts.Count;

        filterInformation = new FilterInformation
        {
            name = name,
            country = country,
            category = category,
            minYear = minYear,
            maxYear = maxYear,
            material = material,
            minWeight = minWeight,
            maxWeight = maxWeight,
            minDiameter = minDiameter,
            maxDiameter = maxDiameter
        };
        sortingInformation = new SortingInformation
        {
            byName = byName,
            byCountry = byCountry,
            byCategory = byCategory,
            byYear = byYear,
            byMaterial = byMaterial,
            byWeight = byWeight,
            byDiameter = byDiameter,
            byDateTime=byDateTime
        };

        allProducts = ApplyFilter(allProducts, filterInformation);
        allProducts = ApplySorting(allProducts, sortingInformation);
        if (filterInformation.name != null)
            allProducts = SearchByName(allProducts, filterInformation.name);

        int pageSize = 50;

        var count = allProducts.Count();
        ViewBag.SearchCount = count;
        var items = allProducts.Skip((page - 1) * pageSize).Take(pageSize).ToList();
        PageViewModel pageViewModel = new PageViewModel(count, page, pageSize);
        CatalogViewModel viewModel = new CatalogViewModel
        {
            PageViewModel = pageViewModel,
            Products = items,
            FilterInformation = filterInformation,
            SortingInformation = sortingInformation
        };
        return View("MyCollection", viewModel);
    }

    public ViewResult MainCatalog(int page = 1, string name = null, string category =
null, string country = null, int? minYear = null, int? maxYear = null, string material =
null,
        float? minWeight = null, float? maxWeight = null, float? minDiameter = null,
float? maxDiameter = null,
        string byName = null, string byYear = null, string byCountry = null, string
byCategory = null, string byMaterial = null, string byWeight = null, string byDiameter =
null,

```

```

        string byDateTime = null)
    {
        ViewBag.UserAuth = User.Identity.IsAuthenticated;
        ViewBag.UserName = User.Identity.Name;
        Array.Sort(RusCountries);
        ViewBag.Countries = RusCountries;
        ViewBag.Materials = RusMaterials;
        ViewBag.Categories = RusCategories;
        User user = db.Users.Include(x=>x.UserProducts).FirstOrDefault(x => x.Email
== User.Identity.Name);

        if (user == null)
            ViewBag.UserRole = "";
        else
            ViewBag.UserRole = user.Role;

        List<Product> allProducts =
db.Products.Include(x=>x.UserAdder).Where(x=>x.isMainCatalog==true).OrderByDescending(x
=> x.DateTime).ToList();
        ViewBag.CatalogCount = allProducts.Count;

        filterInformation = new FilterInformation
        {
            name = name,
            country = country,
            category = category,
            minYear = minYear,
            maxYear = maxYear,
            material = material,
            minWeight = minWeight,
            maxWeight = maxWeight,
            minDiameter = minDiameter,
            maxDiameter = maxDiameter
        };
        sortingInformation = new SortingInformation
        {
            byName = byName,
            byCountry = byCountry,
            byCategory = byCategory,
            byYear = byYear,
            byMaterial = byMaterial,
            byWeight = byWeight,
            byDiameter = byDiameter,
            byDateTime = byDateTime
        };

        allProducts = ApplyFilter(allProducts, filterInformation);
        allProducts = ApplySorting(allProducts, sortingInformation);
        if (filterInformation.name != null)
            allProducts = SearchByName(allProducts, filterInformation.name);

        int pageSize = 50;

        var count = allProducts.Count();
        ViewBag.SearchCount = count;
        var items = allProducts.Skip((page - 1) * pageSize).Take(pageSize).ToList();
        PageViewModel pageViewModel = new PageViewModel(count, page, pageSize);
        CatalogViewModel viewModel = new CatalogViewModel
        {
            PageViewModel = pageViewModel,
            Products = items,

```

```

        FilterInformation = filterInformation,
        SortingInformation = sortingInformation
    };
    return View("MainCatalog", viewModel);
}

[HttpGet]
public ViewResult UserCatalog(int page=1, string name=null, string category=null,
string country = null, int? minYear = null, int? maxYear = null, string material = null,
float? minWeight = null, float? maxWeight = null, float? minDiameter = null,
float? maxDiameter = null,
string byName = null, string byYear = null, string byCountry = null, string
byCategory=null, string byMaterial = null, string byWeight = null, string byDiameter =
null,
string byDateTime = null)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    Array.Sort(RusCountries);
    ViewBag.Countries = RusCountries;
    ViewBag.Materials = RusMaterials;
    ViewBag.Categories = RusCategories;

    List<Product> allProducts =
db.Products.Include(c=>c.UserAdder).Where(x=>x.isMainCatalog==false).OrderByDescending(x
=> x.DateTime).ToList();
    ViewBag.CatalogCount = allProducts.Count;
    filterInformation = new FilterInformation { name = name, country = country,
category = category, minYear = minYear, maxYear = maxYear,
material = material, minWeight = minWeight, maxWeight = maxWeight,
minDiameter = minDiameter, maxDiameter = maxDiameter };
    sortingInformation = new SortingInformation { byName = byName, byCountry =
byCountry, byCategory = byCategory, byYear = byYear,
byMaterial = byMaterial, byWeight = byWeight, byDiameter = byDiameter,
byDateTime = byDateTime
};

    allProducts = ApplyFilter(allProducts, filterInformation);
    allProducts = ApplySorting(allProducts, sortingInformation);
    if (filterInformation.name != null)
        allProducts = SearchByName(allProducts, filterInformation.name);

    int pageSize = 50;

    var count = allProducts.Count();
    ViewBag.SearchCount = count;
    var items = allProducts.Skip((page - 1) * pageSize).Take(pageSize).ToList();
    PageViewModel pageViewModel = new PageViewModel(count, page, pageSize);
    CatalogViewModel viewModel = new CatalogViewModel
    {
        PageViewModel = pageViewModel,
        Products = items,
        FilterInformation = filterInformation,
        SortingInformation = sortingInformation
    };
    return View("UserCatalog", viewModel);
}

public ViewResult AboutUser (int userId, int page = 1, string name = null, string
category = null, string country = null, int? minYear = null, int? maxYear = null, string
material = null,

```

```

        float? minWeight = null, float? maxWeight = null, float? minDiameter = null,
float? maxDiameter = null,
        string byName = null, string byYear = null, string byCountry = null, string
byCategory = null, string byMaterial = null, string byWeight = null, string byDiameter =
null,
        string byDateTime = null)
    {
        ViewBag.UserAuth = User.Identity.IsAuthenticated;
        ViewBag.UserName = User.Identity.Name;
        Array.Sort(RusCountries);
        ViewBag.Countries = RusCountries;
        ViewBag.Materials = RusMaterials;
        ViewBag.Categories = RusCategories;
        User userAbout = db.Users.Include(c => c.UserProducts).ThenInclude(cs =>
cs.Product).FirstOrDefault(x => x.UserId == userId);

        User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);
        if (user == null)
            ViewBag.UserRole = "";
        else
            ViewBag.UserRole = user.Role;

        List<Product> allProducts = userAbout.UserProducts.Select(sc =>
sc.Product).OrderByDescending(x=>x.DateTime).ToList();
        ViewBag.CatalogCount = allProducts.Count;

        filterInformation = new FilterInformation
        {
            name = name,
            country = country,
            category = category,
            minYear = minYear,
            maxYear = maxYear,
            material = material,
            minWeight = minWeight,
            maxWeight = maxWeight,
            minDiameter = minDiameter,
            maxDiameter = maxDiameter
        };
        sortingInformation = new SortingInformation
        {
            byName = byName,
            byCountry = byCountry,
            byCategory = byCategory,
            byYear = byYear,
            byMaterial = byMaterial,
            byWeight = byWeight,
            byDiameter = byDiameter,
            byDateTime = byDateTime
        };

        allProducts = ApplyFilter(allProducts, filterInformation);
        allProducts = ApplySorting(allProducts, sortingInformation);
        if (filterInformation.name != null)
            allProducts = SearchByName(allProducts, filterInformation.name);

        int pageSize = 50;

        var count = allProducts.Count();
        ViewBag.SearchCount = count;
        var items = allProducts.Skip((page - 1) * pageSize).Take(pageSize).ToList();
        PageViewModel pageViewModel = new PageViewModel(count, page, pageSize);

```



```

        CatalogViewModel viewModel = new CatalogViewModel
        {
            PageViewModel = pageViewModel,
            Products = items,
            FilterInformation = filterInformation,
            SortingInformation = sortingInformation,
            User = userAbout
        };
        return View("AboutUser", viewModel);
    }

    private List<Product> ApplyFilter(List<Product> products, FilterInformation
filter)
    {
        int? minYear = filter.minYear;
        int? maxYear = filter.maxYear;
        float? minDiameter = filter.minDiameter;
        float? maxDiameter = filter.maxDiameter;
        float? minWeight = filter.minWeight;
        float? maxWeight = filter.maxWeight;
        string country = filter.country;
        string category = filter.category;
        string material = filter.material;

        if (minYear == null)
            minYear = 0;
        if (maxYear == null)
            maxYear = 10000;
        products = products.Where(x => x.Year <= maxYear && x.Year >=
minYear).ToList();

        if (minWeight == null)
            minWeight = 0;
        if (maxWeight == null)
            maxWeight = 10000;
        products = products.Where(x => x.Weight <= maxWeight && x.Weight >=
minWeight).ToList();

        if (minDiameter == null)
            minDiameter = 0;
        if (maxDiameter == null)
            maxDiameter = 10000;
        products = products.Where(x => x.Diameter >= minDiameter && x.Year <=
maxDiameter).ToList();

        if (country != null)
            products = products.Where(x => x.Country == country).ToList();
        if (category != null)
            products = products.Where(x => x.Category == category).ToList();
        if (material != null)
            products = products.Where(x => x.Material == material).ToList();

        return products;
    }
    private List<Product> ApplySorting(List<Product> products, SortingInformation
sorter)
    {
        if (sorter.byDiameter == "По возрастанию")
            products = products.OrderBy(x => x.Diameter).ToList();
        else if (sorter.byDiameter == "По убыванию")
            products = products.OrderByDescending(x => x.Diameter).ToList();
    }

```

					ДП ІС-6210.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (sorter.byWeight == "По возрастанию")
            products = products.OrderBy(x => x.Weight).ToList();
        else if (sorter.byWeight == "По убыванию")
            products = products.OrderByDescending(x => x.Weight).ToList();

        if (sorter.byMaterial == "По возрастанию")
            products = products.OrderBy(x => x.Material).ToList();
        else if (sorter.byMaterial == "По убыванию")
            products = products.OrderByDescending(x => x.Material).ToList();

        if (sorter.byYear == "По возрастанию")
            products = products.OrderBy(x => x.Year).ToList();
        else if (sorter.byYear == "По убыванию")
            products = products.OrderByDescending(x => x.Year).ToList();

        if (sorter.byCategory == "По возрастанию")
            products = products.OrderBy(x => x.Category).ToList();
        else if (sorter.byCategory == "По убыванию")
            products = products.OrderByDescending(x => x.Category).ToList();

        if (sorter.byCountry == "По возрастанию")
            products = products.OrderBy(x => x.Country).ToList();
        else if (sorter.byCountry == "По убыванию")
            products = products.OrderByDescending(x => x.Country).ToList();

        if (sorter.byName == "По возрастанию")
            products = products.OrderBy(x => x.Name).ToList();
        else if (sorter.byName == "По убыванию")
            products = products.OrderByDescending(x => x.Name).ToList();

        if (sorter.byDateTime == "По возрастанию")
            products = products.OrderBy(x => x.DateTime).ToList();
        else if (sorter.byDateTime == "По убыванию")
            products = products.OrderByDescending(x => x.DateTime).ToList();

        return products;
    }

```

```

public IActionResult ChangeRoleUser(int userId, string newRole)
{
    string curName = User.Identity.Name;
    User curUser = db.Users.FirstOrDefault(x => x.Email == curName);
    User user = db.Users.FirstOrDefault(x => x.UserId == userId);
    if (curUser.Role == "Адміністратор" && user.Role != "Адміністратор")
    {

        user.Role = newRole;
        db.Update(user);
        db.SaveChanges();
    }
    return RedirectToAction("AboutUser", new { userId = userId });
}

```

```

public ViewResult UserSearch(string userName)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    List<User> users = db.Users.Where(x =>
x.Login.ToLower().Contains(userName.ToLower())).ToList();

```

```

        return View(users);
    }

    public List<Product> SearchByName(List<Product> products, string name)
    {
        List<ProductWithStringLength> allProductsWSL = new
List<ProductWithStringLength>();

        foreach (Product pr in products)
        {
            int x = MaxHalfString(pr.Name.ToLower(), name.ToLower());
            if ((x * 100 / name.Length > 20) && (x!=1))
                allProductsWSL.Add(new ProductWithStringLength { Product = pr,
StringLength = x });
        }
        allProductsWSL = allProductsWSL.OrderByDescending(x =>
x.StringLength).ToList();
        List<Product> myProducts = new List<Product>();

        foreach(ProductWithStringLength pr in allProductsWSL)
        {
            myProducts.Add(pr.Product);
        }
        return myProducts;
    }

    public int MaxHalfString(string s1, string s2)
    {
        char[] s1Symbols = s1.ToCharArray();
        char[] s2Symbols = s2.ToCharArray();
        int max = 0;
        int[,] A = new int[s1Symbols.Length + 1, s2Symbols.Length + 1];
        for (int i=0; i<s1Symbols.Length+1; i++)
        {
            for (int j=0; j<s2Symbols.Length+1; j++)
            {
                if((i==0) || (j==0))
                {
                    A[i, j] = 0;
                }
                if((i!=0) && (j!=0))
                {
                    if (s1Symbols[i-1] != s2Symbols[j-1])
                        A[i, j] = 0;
                    else
                        A[i, j] = A[i - 1, j - 1] + 1;
                }
                if (max < A[i, j])
                    max = A[i, j];
            }
        }
        return max;
    }
}

```

ProductController.cs

```

using MedalsWebSystem.Models;
using MedalsWebSystem.ViewModels;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Threading.Tasks;
using static System.Net.Mime.MediaTypeNames;

namespace MedalsWebSystem.Controllers
{
    public class ProductController : Controller
    {
        private ApplicationDbContext db;
        public ProductController(ApplicationContext context)
        {
            db = context;
        }

        public string[] RusCountries { get; } = new string[] { "Китай", "Андорра",
"Аргентина", "Австрія", "Білорусь", "Бельгія", "Болгарія",
"Канада", "Хорватія", "Куба", "Кіпр", "Чеська Республіка", "Данія", "Еквадор",
"Естонія", "Фінляндія", "Франція",
"Німеччина", "Греція", "Гватемала", "Угорщина", "Ірландія", "Острів Мен",
"Італія", "Японія", "Казахстан", "Латвія",
"Литва", "Люксембург", "Македонія", "Мальта", "Мексика", "Молдавія", "Монако",
"Нідерланди", "Нікарагуа", "Норвегія",
"Парагвай", "Перу", "Польща", "Португалія", "Румунія", "Росія", "Сан Марино",
"Сербія", "Словаччина", "Словенія",
"Південна Корея", "Іспанія", "Швеція", "Швейцарія", "Туреччина", "Україна",
"Великобританія", "США", "Ватикан"};

        public string[] RusMaterials { get; } = new string[] { "Золото", "Серебро",
"Мідь", "Бронза", "Бронза", "Томпак", "Алюміній",
"Меднонікель", "Нейзильбер", "Сталь", "Біметал", "Інше"};

        public string[] RusCategories { get; } = new string[] { "Архітектура",
"Мистецтво", "Спорт", "Наука", "Промисловість",
"Транспорт", "Фауна", "Флора", "Освоєння космосу", "Історія та особистості",
"Інше"};

        [HttpGet]
        [Authorize]
        public IActionResult AddProduct()
        {
            ViewBag.UserAuth = User.Identity.IsAuthenticated;
            ViewBag.UserName = User.Identity.Name;
            Array.Sort(RusCountries);
            ViewBag.Countries = RusCountries;
            ViewBag.Materials = RusMaterials;
            ViewBag.Categories = RusCategories;

            return View();
        }
    }
}

```

					ДП ІС-6210.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

[HttpPost]
[Authorize]
public IActionResult AddProduct(AddProductModel product)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    Array.Sort(RusCountries);
    ViewBag.Countries = RusCountries;
    ViewBag.Materials = RusMaterials;
    ViewBag.Categories = RusCategories;

    User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);

    if (product.Photo != null && product.Photo.Length >= 5242880)
    {
        ModelState.AddModelError("Photo", "Розмір файлу повинен бути менше 5мб");
    }
    if (ModelState.IsValid)
    {
        Product newPr = new Product
        {
            Name = product.Name,
            Country = product.Country,
            Category = product.Category,
            Year = product.Year,
            Material = product.Material,
            Weight = product.Weight,
            Diameter = product.Diameter,
            Description = product.Description,
            UserAdder = user,
            DateTime = DateTime.UtcNow
        };
        if (product.Photo != null)
        {
            byte[] imageData = null;
            byte[] minImageData = null;

            using (var binaryReader = new
BinaryReader(product.Photo.OpenReadStream()))
            {
                imageData = binaryReader.ReadBytes((int)product.Photo.Length);
            }

            using (MemoryStream ms = new MemoryStream(imageData, 0,
imageData.Length))
            {
                using (System.Drawing.Image img =
System.Drawing.Image.FromStream(ms))
                {
                    int h = 75;
                    int w = 150;

                    using (Bitmap b = new Bitmap(img, new Size(w, h)))
                    {
                        using (MemoryStream ms2 = new MemoryStream())
                        {
                            b.Save(ms2, System.Drawing.Imaging.ImageFormat.Jpeg);
                            minImageData = ms2.ToArray();
                        }
                    }
                }
            }
        }
    }
}

```

```

    }

    newPr.Photo = imageData;
    newPr.minPhoto = minImageData;
}
db.Add(newPr);
db.SaveChanges();
user.UserProducts.Add(new UserProduct { ProductId = newPr.ProductId,
UserId = user.UserId });
db.SaveChanges();
return RedirectToAction("MyCollection", "Home");
}
return View();
}

[HttpGet]
[Authorize]
public ViewResult UpdateProduct(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    Array.Sort(RusCountries);
    ViewBag.Countries = RusCountries;
    ViewBag.Materials = RusMaterials;
    ViewBag.Categories = RusCategories;

    Product pr =
db.Products.Include(u=>u.UserProducts).ThenInclude(uc=>uc.User).FirstOrDefault(x =>
x.ProductId == productId);
    List<User> us = pr.UserProducts.Select(uc => uc.User).ToList();

    AddProductModel prView = new AddProductModel
    {
        ProductId = pr.ProductId,
        Name = pr.Name,
        Country = pr.Country,
        Category = pr.Category,
        Year = pr.Year,
        Material = pr.Material,
        Weight = pr.Weight,
        Diameter = pr.Diameter,
        Description = pr.Description
    };

    bool key=false;
    foreach (User user in us)
    {
        if (user.Email == ViewBag.UserName)
            key = true;
    }
    if ((key) && (us != null))
    {
        return View(prView);
    }
    else
        return View("AddProduct");
}

[HttpPost]
public IActionResult UpdateProduct(AddProductModel product)
{

```

```

        ViewBag.UserAuth = User.Identity.IsAuthenticated;
        ViewBag.UserName = User.Identity.Name;
        Array.Sort(RusCountries);
        ViewBag.Countries = RusCountries;
        ViewBag.Materials = RusMaterials;
        ViewBag.Categories = RusCategories;
        if (product.Photo != null && product.Photo.Length >= 5242880)
        {
            ModelState.AddModelError("Photo", "Розмір файлу повинен бути менше 5мб");
        }
        if (ModelState.IsValid)
        {
            Product pr = db.Products.FirstOrDefault(x => x.ProductId ==
product.ProductId);
            pr.Name = product.Name;
            pr.Year = product.Year;
            pr.Country = product.Country;
            pr.Category = product.Category;
            pr.Material = product.Material;
            pr.Weight = product.Weight;
            pr.Diameter = product.Diameter;
            pr.Description = product.Description;
            if (product.Photo != null)
            {
                byte[] imageData = null;
                byte[] minImageData = null;

                using (var binaryReader = new
BinaryReader(product.Photo.OpenReadStream()))
                {
                    imageData = binaryReader.ReadBytes((int)product.Photo.Length);
                }

                using (MemoryStream ms = new MemoryStream(imageData, 0,
imageData.Length))
                {
                    using (System.Drawing.Image img =
System.Drawing.Image.FromStream(ms))
                    {
                        int h = 75;
                        int w = 150;

                        using (Bitmap b = new Bitmap(img, new Size(w, h)))
                        {
                            using (MemoryStream ms2 = new MemoryStream())
                            {
                                b.Save(ms2, System.Drawing.Imaging.ImageFormat.Jpeg);
                                minImageData = ms2.ToArray();
                            }
                        }
                    }
                }

                pr.Photo = imageData;
                pr.minPhoto = minImageData;
            }
            db.Update(pr);
            db.SaveChanges();
            return RedirectToAction("MyCollection", "Home");
        }
        return View(product);
    }

```

```

[Authorize]
public ActionResult DeleteProduct(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    Product product = db.Products.Include(u => u.UserProducts).ThenInclude(uc =>
uc.User).FirstOrDefault(x => x.ProductId == productId);
    return View(product);
}

public IActionResult DeleteProductAction(int productId)
{
    ViewBag.UserName = User.Identity.Name;
    Product product = db.Products.Include(u => u.UserAdder).FirstOrDefault(x =>
x.ProductId == productId);
    User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);
    if ((product.UserAdder.Email == User.Identity.Name &&
product.isMainCatalog==false)
        || user.Role == "Адміністратор" || user.Role == "Модератор")
    {
        db.Products.Remove(product);
        db.SaveChanges();
    }
    return RedirectToAction("MyCollection", "Home");
}

public ActionResult ProductInfo(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    User user =
db.Users.Include(xc=>xc.UserProducts).ThenInclude(xc=>xc.Product).FirstOrDefault(x =>
x.Email == User.Identity.Name);
    ViewBag.UserRole = user.Role;
    List<Product> curUserProducts = user.UserProducts.Select(x =>
x.Product).ToList();

    ViewBag.CanIAddIt = true;
    ViewBag.CanIDeleteIt = false;

    foreach (Product pr in curUserProducts)
    {
        if (pr.ProductId == productId)
        {
            ViewBag.CanIAddIt = false;
            ViewBag.CanIDeleteIt = true;
        }
    }

    Product product = db.Products.Include(x => x.UserProducts).ThenInclude(xs =>
xs.User).FirstOrDefault(x => x.ProductId == productId);
    if (product.isMainCatalog==false)
    {
        ViewBag.CanIAddIt = false;
        ViewBag.CanIDeleteIt = false;
    }

    List<User> users = product.UserProducts.Select(x => x.User).ToList();
    ProductInfoViewModel viewModel = new ProductInfoViewModel
    {

```



```

        Product = product,
        Users = users
    };
    return View(viewModel);
}

[Authorize]
public IActionResult AddToMyCollection(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    Product product = db.Products.FirstOrDefault(x => x.ProductId == productId);
    User user =
db.Users.Include(x=>x.UserProducts).ThenInclude(xc=>xc.Product).FirstOrDefault(x =>
x.Email == User.Identity.Name);
    bool key = false;
    foreach( Product pr in user.UserProducts.Select(x=>x.Product))
    {
        if (pr.ProductId==productId)
        {
            key = true;
        }
    }
    if(!key && product.isMainCatalog)
    {
        user.UserProducts.Add(new UserProduct { UserId = user.UserId, ProductId =
productId });
        db.SaveChanges();
    }

    return RedirectToAction("MyCollection", "Home");
}

[Authorize]
public IActionResult DeleteFromMyCollection(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    User user = db.Users.Include(x => x.UserProducts).ThenInclude(xc =>
xc.Product).FirstOrDefault(x => x.Email == User.Identity.Name);
    Product product = db.Products.FirstOrDefault(x => x.ProductId == productId);

    bool key = false;
    foreach (Product pr in user.UserProducts.Select(x => x.Product))
    {
        if (pr.ProductId == productId)
        {
            key = true;
        }
    }
    if (key && product.isMainCatalog)
    {
        var userProduct = user.UserProducts.FirstOrDefault(x => x.ProductId ==
productId);
        user.UserProducts.Remove(userProduct);
        db.SaveChanges();
    }

    return RedirectToAction("MyCollection", "Home");
}

[Authorize]

```

```

public IActionResult AddToMainCatalog(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    User user = db.Users.Include(x => x.UserProducts).ThenInclude(xc =>
xc.Product).FirstOrDefault(x => x.Email == User.Identity.Name);
    Product product = db.Products.FirstOrDefault(x => x.ProductId == productId);

    if (!product.isMainCatalog && (user.Role == "Адміністратор" || user.Role
=="Модератор"))
    {
        product.isMainCatalog = true;
        db.Products.Update(product);
        db.SaveChanges();
    }
    return RedirectToAction("MainCatalog", "Home");
}

[Authorize]
public IActionResult DeleteFromMainCatalog(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    User user = db.Users.Include(x => x.UserProducts).ThenInclude(xc =>
xc.Product).FirstOrDefault(x => x.Email == User.Identity.Name);
    Product product = db.Products.FirstOrDefault(x => x.ProductId == productId);

    if (product.isMainCatalog && (user.Role == "Адміністратор" || user.Role ==
"Модератор"))
    {
        product.isMainCatalog = false;
        db.Products.Update(product);
        db.SaveChanges();
    }
    return RedirectToAction("MainCatalog", "Home");
}

[HttpGet]
[Authorize]
public ViewResult AddProductMC()
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);
    Array.Sort(RusCountries);
    ViewBag.Countries = RusCountries;
    ViewBag.Materials = RusMaterials;
    ViewBag.Categories = RusCategories;

    if (user.Role == "Адміністратор" || user.Role == "Модератор")
    {
        return View();
    }
    else return View("MainCatalog", "Home");
}

[HttpPost]
[Authorize]
public IActionResult AddProductMC(AddProductModel product)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;

```

```

User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);
ViewBag.Countries = RusCountries;
ViewBag.Materials = RusMaterials;
ViewBag.Categories = RusCategories;

if (product.Photo != null && product.Photo.Length >= 5242880)
{
    ModelState.AddModelError("Photo", "Розмір файлу повинен бути менше 5мб");
}
if (ModelState.IsValid)
{
    if (user.Role == "Адміністратор" || user.Role == "Модератор")
    {
        Product newPr = new Product
        {
            Name = product.Name,
            Country = product.Country,
            Category = product.Category,
            Year = product.Year,
            Material = product.Material,
            Weight = product.Weight,
            Diameter = product.Diameter,
            Description = product.Description,
            UserAdder = user,
            DateTime = DateTime.UtcNow,
            isMainCatalog = true
        };
        if (product.Photo != null)
        {
            byte[] imageData = null;
            byte[] minImageData = null;

            using (var binaryReader = new
BinaryReader(product.Photo.OpenReadStream()))
            {
                imageData =
binaryReader.ReadBytes((int)product.Photo.Length);
            }

            using (MemoryStream ms = new MemoryStream(imageData, 0,
imageData.Length))
            {
                using (System.Drawing.Image img =
System.Drawing.Image.FromStream(ms))
                {
                    int h = 75;
                    int w = 150;

                    using (Bitmap b = new Bitmap(img, new Size(w, h)))
                    {
                        using (MemoryStream ms2 = new MemoryStream())
                        {
                            b.Save(ms2,
System.Drawing.Imaging.ImageFormat.Jpeg);
                            minImageData = ms2.ToArray();
                        }
                    }
                }
            }

            newPr.Photo = imageData;
            newPr.minPhoto = minImageData;

```

```

        }
        db.Add(newPr);
        db.SaveChanges();

    }

    return RedirectToAction("MainCatalog", "Home");
}
return View();
}

[HttpGet]
[Authorize]
public ViewResult UpdateProductMC(int productId)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    Array.Sort(RusCountries);
    ViewBag.Countries = RusCountries;
    ViewBag.Materials = RusMaterials;
    ViewBag.Categories = RusCategories;

    User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);
    Product pr = db.Products.Include(u => u.UserProducts).ThenInclude(uc =>
uc.User).FirstOrDefault(x => x.ProductId == productId);

    AddProductModel prView = new AddProductModel
    {
        ProductId = pr.ProductId,
        Name = pr.Name,
        Country = pr.Country,
        Category = pr.Category,
        Year = pr.Year,
        Material = pr.Material,
        Weight = pr.Weight,
        Diameter = pr.Diameter,
        Description = pr.Description
    };

    if ((user.Role == "Адміністратор" || user.Role == "Модератор") &&
pr.isMainCatalog)
    {
        return View("UpdateProductMC", prView);
    }
    else
        return View();
}

[HttpPost]
[Authorize]
public IActionResult UpdateProductMC(AddProductModel product)
{
    ViewBag.UserAuth = User.Identity.IsAuthenticated;
    ViewBag.UserName = User.Identity.Name;
    User user = db.Users.FirstOrDefault(x => x.Email == User.Identity.Name);
    ViewBag.Countries = RusCountries;
    ViewBag.Materials = RusMaterials;
    ViewBag.Categories = RusCategories;

    if (product.Photo != null && product.Photo.Length >= 5242880)
    {
        ModelState.AddModelError("Photo", "Розмір файлу повинен бути менше 5мб");
    }
}

```

```

    }
    if (ModelState.IsValid)
    {
        if ((user.Role == "Адміністратор" || user.Role == "Модератор"))
        {
            Product pr = db.Products.FirstOrDefault(x => x.ProductId ==
product.ProductId);
            pr.Name = product.Name;
            pr.Year = product.Year;
            pr.Country = product.Country;
            pr.Category = product.Category;
            pr.Material = product.Material;
            pr.Weight = product.Weight;
            pr.Diameter = product.Diameter;
            pr.Description = product.Description;
            if (product.Photo != null)
            {
                byte[] imageData = null;
                byte[] minImageData = null;

                using (var binaryReader = new
BinaryReader(product.Photo.OpenReadStream()))
                {
                    imageData =
binaryReader.ReadBytes((int)product.Photo.Length);
                }

                using (MemoryStream ms = new MemoryStream(imageData, 0,
imageData.Length))
                {
                    using (System.Drawing.Image img =
System.Drawing.Image.FromStream(ms))
                    {
                        int h = 75;
                        int w = 150;

                        using (Bitmap b = new Bitmap(img, new Size(w, h)))
                        {
                            using (MemoryStream ms2 = new MemoryStream())
                            {
                                b.Save(ms2,
System.Drawing.Imaging.ImageFormat.Jpeg);
                                minImageData = ms2.ToArray();
                            }
                        }
                    }
                }
                pr.Photo = imageData;
                pr.minPhoto = minImageData;
            }
            db.Update(pr);
            db.SaveChanges();
        }
        return RedirectToAction("MainCatalog", "Home");
    }
    return View(product);
}
}
}

```

ApplicationContext.cs

					ДП ІС-6210.00.000 ПЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.Models
{
    public class ApplicationContext : DbContext
    {
        public DbSet<User> Users { get; set; }
        public DbSet<Product> Products { get; set; }
        public DbSet<Dialogue> Dialogues { get; set; }
        public DbSet<Message> Messages { get; set; }
        public ApplicationContext(DbContextOptions<ApplicationContext> options)
            : base(options)
        {
        }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<UserDialogue>()
                .HasKey(t => new { t.UserId, t.DialogueId });

            modelBuilder.Entity<UserDialogue>()
                .HasOne(sc => sc.User)
                .WithMany(s => s.UserDialogues)
                .HasForeignKey(sc => sc.UserId);

            modelBuilder.Entity<UserDialogue>()
                .HasOne(sc => sc.Dialogue)
                .WithMany(c => c.UserDialogues)
                .HasForeignKey(sc => sc.DialogueId);

            modelBuilder.Entity<UserProduct>()
                .HasKey(t => new { t.UserId, t.ProductId });

            modelBuilder.Entity<UserProduct>()
                .HasOne(sc => sc.User)
                .WithMany(s => s.UserProducts)
                .HasForeignKey(sc => sc.UserId);

            modelBuilder.Entity<UserProduct>()
                .HasOne(sc => sc.Product)
                .WithMany(c => c.UserProducts)
                .HasForeignKey(sc => sc.ProductId);
        }
    }
}

```

Dialogue.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.Models
{

```

```

public class Dialogue
{
    public int DialogueId { get; set; }
    public List<Message> Messages { get; set; }
    public List<UserDialogue> UserDialogues { get; set; }

    public Dialogue()
    {
        UserDialogues = new List<UserDialogue>();
    }
}

```

Message.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.Models
{
    public class Message
    {
        public int MessageId { get; set; }
        public string Text { get; set; }
        public DateTime Time { get; set; }
        public User Sender { get; set; }
        public int DialogueId { get; set; }
        public Dialogue Dialogue { get; set; }
    }
}

```

Product.cs

```

using Microsoft.AspNetCore.Http;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.Models
{
    public class Product
    {
        public int ProductId { get; set; }
        public string Name { get; set; }
        public string Category { get; set; }
        public string Country { get; set; }
        public int? Year { get; set; }
        public string Material { get; set; }
        public float? Diameter { get; set; }
        public float? Weight { get; set; }
        public string Description { get; set; }
        public byte[] Photo { get; set; }
        public byte[] minPhoto { get; set; }
        public bool isMainCatalog { get; set; }
        public DateTime DateTime { get; set; }
        public User UserAdder { get; set; }
    }
}

```

```

        public List<UserProduct> UserProducts { get; set; }
        public Product()
        {
            UserProducts = new List<UserProduct>();
        }
    }
}

```

User.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.Models
{
    public class User
    {
        public int UserId { get; set; }
        public string Name { get; set; }
        public string Surname { get; set; }
        public string Login { get; set; }
        public string Email { get; set; }
        public string Pass { get; set; }
        public string Role { get; set; }
        public List<UserProduct> UserProducts { get; set; }
        public List<UserDialogue> UserDialogues { get; set; }
        public User()
        {
            UserProducts = new List<UserProduct>();
            UserDialogues = new List<UserDialogue>();
        }
    }
}

```

UserDialogue.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.Models
{
    public class UserDialogue
    {
        public int UserId { get; set; }
        public User User { get; set; }
        public bool IsRead { get; set; }
        public int DialogueId { get; set; }
        public Dialogue Dialogue { get; set; }
    }
}

```

UserProduct.cs

```

using System;

```



```

using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.Models
{
    public class UserProduct
    {
        public int UserId { get; set; }
        public User User { get; set; }
        public int ProductId { get; set; }
        public Product Product { get; set; }
    }
}

```

AddProductModel.cs

```

using Microsoft.AspNetCore.Http;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class AddProductModel
    {
        public int ProductId { get; set; }
        [Required(ErrorMessage = "Не вказано назву")]
        [StringLength(50, MinimumLength = 1, ErrorMessage = "Назва повинна бути не більше 50 символів")]
        public string Name { get; set; }
        [Required(ErrorMessage = "Не вказана категория")]
        public string Category { get; set; }
        [Required(ErrorMessage = "Не вказана категорія")]
        public string Country { get; set; }
        [Required(ErrorMessage = "Не вказаний рік")]
        [Range(0, 3000, ErrorMessage = "Недопустиме значення")]
        public int? Year { get; set; }
        [Required(ErrorMessage = "Не вказаний діаметр")]
        [Range(0, 3000, ErrorMessage = "Недопустиме значення")]
        public float? Diameter { get; set; }
        [Required(ErrorMessage = "Не вказана вага")]
        [Range(0, 100000, ErrorMessage = "Недопустиме значення")]
        public float? Weight { get; set; }
        [Required(ErrorMessage = "Не вказаний матеріал")]
        public string Material { get; set; }
        public string Description { get; set; }
        public IFormFile Photo { get; set; }
    }
}

```

CatalogueViewModel.cs

```

using MedalsWebSystem.Models;
using System;
using System.Collections.Generic;
using System.Linq;

```

					ДП ІС-6210.00.000 ПЗ	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class CatalogViewModel
    {
        public IEnumerable<Product> Products { get; set; }
        public PageViewModel PageViewModel { get; set; }
        public FilterInformation FilterInformation { get; set; }
        public SortingInformation SortingInformation { get; set; }

        public User User { get; set; } = null;
    }
}

```

DialogueViewModel.cs

```

using MedalsWebSystem.Models;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class DialogueViewModel
    {
        public Dialogue Dialogue { get; set; }
        public User User2 { get; set; }
        public List<Message> Messages { get; set; }
        public PageViewModel PageViewModel { get; set; }
        public NewMessageViewModel NewMessageViewModel { get; set; }
    }
}

```

FilterInformation.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class FilterInformation
    {
        public string name { get; set; }
        public string category { get; set; }
        public int? minYear { get; set; }
        public int? maxYear { get; set; }
        public float? minWeight { get; set; }
        public float? maxWeight { get; set; }
        public float? minDiameter { get; set; }
        public float? maxDiameter { get; set; }
        public string country { get; set; }
        public string material { get; set; }
    }
}

```

LoginModel.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class LoginModel
    {
        [Required(ErrorMessage = "Не вказан Email")]
        public string Email { get; set; }

        [Required(ErrorMessage = "Не вказан пароль")]
        [DataType(DataType.Password)]
        public string Password { get; set; }
    }
}

```

MyDialoguesViewModel.cs

```

using MedalsWebSystem.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class MyDialoguesViewModel
    {
    }

    public class UserAndRead
    {
        public User User { get; set; }
        public bool IsRead { get; set; }
    }
}

```

NewMessageViewModel.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class NewMessageViewModel
    {
        [Required(ErrorMessage = "Повідомлення порожнє")]
        [StringLength(250, MinimumLength = 1, ErrorMessage = "Повідомлення повинно бути не більше 250 символів")]
    }
}

```

```

        public string Text;

        public int dialogueId;
        public int user2Id;
    }
}

```

PageViewModel.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class PageViewModel
    {
        public int PageNumber { get; private set; }
        public int TotalPages { get; private set; }
        public PageViewModel(int count, int page, int pageSize)
        {
            PageNumber = page;
            TotalPages = (int)Math.Ceiling(count / (double)pageSize);
        }
    }
}

```

ProductInfoModel.cs

```

using MedalsWebSystem.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class ProductInfoViewModel
    {
        public List<User> Users { get; set; }
        public Product Product { get; set; }
    }
}

```

ProductWithStringLength.cs

```

using MedalsWebSystem.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class ProductWithStringLength
    {
        public Product Product { get; set; }
    }
}

```

```

        public int StringLength { get; set; }
    }
}

```

RegisterModel.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class RegisterModel
    {
        [Required(ErrorMessage = "Не вказаний Email")]
        [RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}",
        ErrorMessage = "Некоректна адреса")]
        public string Email { get; set; }

        [Required(ErrorMessage = "Не вказане ім'я")]
        public string Name { get; set; }

        [Required(ErrorMessage = "Не вказане прізвище")]
        public string Surname { get; set; }

        [Required(ErrorMessage = "Не вказаний логин")]
        public string Login { get; set; }

        [Required(ErrorMessage = "Не вказаний пароль")]
        [DataType(DataType.Password)]
        public string Password { get; set; }

        [DataType(DataType.Password)]
        [Compare("Password", ErrorMessage = "Пароль введено невірно")]
        public string ConfirmPassword { get; set; }
    }
}

```

SortingInformation.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace MedalsWebSystem.ViewModels
{
    public class SortingInformation
    {
        public string byName { get; set; }
        public string byCountry { get; set; }
        public string byCategory { get; set; }
        public string byYear { get; set; }
        public string byMaterial { get; set; }
        public string byWeight { get; set; }
        public string byDiameter { get; set; }
        public string byDateTime { get; set; }
    }
}

```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проекту

_____ Оксана ЖУРАКОВСЬКА

(підпис)

(ініціали, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

(підпис)

(ініціали, прізвище)

“14” квітня 2020 р.

«Інформаційна система підтримки діяльності фалеристів»

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр ДП ІС-6210.01.000 ТЗ

на 8 сторінках

Київ – 2020 року

Зміст

Зміст	2
1. Загальні положення	3
1.1 Повне найменування системи та її умовне позначення	3
1.2 Найменування організації-замовника та організацій-учасників робіт	3
1.3 Перелік документів, на підставі яких створюється система	3
1.4 Планові терміни початку і закінчення роботи зі створення системи	3
2. Призначення і цілі створення системи	3
2.1 Призначення системи	3
2.2 Цілі створення системи	3
3. Характеристика об'єкта автоматизації	4
4. Вимоги до програмного забезпечення	5
4.1 Вимоги до функціональних характеристик	5
4.2 Вимоги до надійності	6
4.3 Вимоги до складу і параметрів технічних засобів	6
5. Стадії і етапи розробки	7
6. Порядок контролю та приймання	8
6.1 Види випробувань	8

					ДП 6210 01.000 ТЗ		
Зм.	Арк.	ПІБ	Підп.	Дата	Технічне завдання		
Розробн.		Ковинев К.О.					
Керівн.		Жураковська О.С.					
Консульт.							
Н/контр.		Новінський В.П.					
В.о.зав.каф.		Павлов О.А.					
					Лит.	Арк.	Аркушів
						2	8
					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		

1. Загальні положення

1.1 Повне найменування системи та її умовне позначення

Повне найменування: *Інформаційна система підтримки діяльності фалеристів.*

Умовне позначення: *Веб-додаток для колекціонерів.*

1.2 Найменування організації-замовника та організацій-учасників робіт

Організація-замовник: *Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», кафедра Автоматизованих систем обробки інформації та управління факультету інформатики та обчислювальної техніки.*
Представник замовника: *доцент кафедри АСОІУ Жураковська Оксана Сергіївна.*

Адреса замовника: *м. Київ проспект Перемоги 37.*

Учасник робіт: *студент групи ІС-62, кафедри АСОІУ, НТУУ «КПІ ім. І. Сікорського» Ковинев Кирило Олексійович.*

1.3 Перелік документів, на підставі яких створюється система

Перелік нормативних документів, на підставі яких створюється система:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Початок роботи: *13.04.2020.*

Закінчення роботи: *14.05.2020.*

2. Призначення і цілі створення системи

2.1 Призначення системи

Призначенням розробки застосування є підтримка діяльності колекціонерів настільних медалей

					ДП ІС-6210.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2.2 Цілі створення системи

Спрощення та підвищення ефективності процесу колекціонування експонування за рахунок розміщення та зберігання електронної версії колекції на сайті.

3. Характеристика об'єкта автоматизації

Веб-застосування повинно забезпечити колекціонерам зручне зберігання та демонстрацію електронних версій різних колекцій настільних медалей іншим колекціонерам.

Система повинна будуватися з урахуванням наступних особливостей:

- тільки авторизований колекціонер може вести та демонструвати свою електронну версію колекцію;
- переглядати каталоги з настільними медалями може будь-який користувач;
- колекціонер може мати декілька настільних медалей;
- настільна медаль може бути у декількох колекціонерів;
- не уся інформація про характеристики медалі може бути відома.

Кожна медаль характеризується наступними параметрами:

- назва;
- країна випуску;
- рік випуску;
- матеріал;
- діаметр;
- вага;
- опис;
- фото.

Для медалей потрібно зберігати інформацію про користувачів, в яких вона є.

Кожен зареєстрований користувач характеризується наступними параметрами:

- прізвище;
- ім'я;
- логін;
- електронна пошта;
- пароль.

Для користувачів потрібно зберігати інформацію про медалі, які у нього є.

Система створюється для наступних груп користувачів:

- адміністрація;
- модератори;

					ДП ІС-6210.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

- звичайні користувачі.

Множина медалей називається «каталогом». Множина електронних версій медалей користувача, які є в його колекції, називаються «каталогом власної колекції». Множина медалей електронних версій, які розміщені на сайті модераторами, називаються «головним\основним каталогом сайту».

Користувачі повинні мати можливість інформаційної підтримки бази даних застосунку: ведення бази даних (запис, читання, модифікація, видалення), забезпечення логічної несуперечливості бази даних.

Користувачі застосунку повинні мати можливість:

- ведення каталогу власної колекції;
- отримати список медалей конкретного каталогу власної колекції користувача або головного каталогу сайту;
- отримати інформацію про каталог;
- отримати інформацію про користувача;
- отримати інформацію про конкретну медаль;
- комунікації з іншими користувачами.

Модератори та адміністрація повинні підтримувати контент сайту у належному вигляді та поповнювати головний каталог сайту.

Модератори застосунку повинні мати можливість:

- видалення медалі з будь-якого каталогу на сайті;
- ведення головного каталогу сайту.

Адміністратори застосунку повинні мати можливість:

- видалення медалі з будь-якого каталогу на сайті;
- ведення головного каталогу сайту;
- зміна ролі/статусу користувача.

Предметом комунікації між користувачами виступає чат з текстовими повідомленнями, кожне з яких має наступні характеристики:

- зміст повідомлення;
- час відправлення;
- відправник;
- отримувач.

4. Вимоги до програмного забезпечення

4.1 Вимоги до функціональних характеристик

Функціональні вимоги системи:

- система надає можливість користувачу створити обліковий запис;
- система надає можливість користувачу авторизуватися;

					ДП ІС-6210.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

- система надає можливість користувачу ведення інформації про окрему медаль;
- система надає можливість користувачу завантажити фото окремої медалі;
- система надає можливість користувачу перегляду інформації про кожну медаль;
- система надає інформацію про медалі за результатом запиту користувача з застосуванням пошуку, фільтрів та сортування;
- система надає користувачу інформацію за запитом про іншого користувача;
- система надає можливість користувачу комунікації з іншими користувачами;
- система надає можливість користувачам зі статусом «Модератор» або «Адміністратор» видаляти будь-яку медаль;
- система надає можливість користувачам зі статусом «Адміністратор» змінювати статус інших користувачів.

4.2 Вимоги до надійності

- 1) Контроль введеної інформації:
 - користувач не може ввести не чисельні символи до чисельних полів;
 - користувач не може ввести занадто довгу інформацію до інформаційних полів;
 - користувач не може завантажити фото більше 5 мегабайт;
 - інформаційні поля для введення інформації не можуть бути пустими.
- 2) Захист від некоректних дій користувача:
 - користувач не може видалити\редагувати медаль, яка йому не належить;
 - користувач не може виконувати функції, якщо його статус йому це не дозволяє.
- 3) Забезпечення цілісності інформації у базі даних:
 - видалення запису медалі з таблиці Products супроводжується видаленням усіх записів з таблиці UserProducts, пов'язаних з медаллю.

4.3 Вимоги до складу і параметрів технічних засобів

Мінімальні системні вимоги:

- а) Процесор: з частотою не менше 1 ГГц;
- б) Обсяг ОЗУ: не менше 64 мегабайт;
- в) Операційна система: Windows XP/7/8/10;
- г) Наявність одного з браузерів:
 - 1) Internet Explorer;
 - 2) Opera;
 - 3) Google Chrome;

					ДП ІС-6210.01.000 ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

- 4) Mozilla Firefox;
5) Microsoft Edge.

5. Стадії і етапи розробки

Стадії і етапи розробки:

Таблиця 5.1 – Календарний план виконання робіт

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
14.	<i>Вивчення рекомендованої літератури</i>	<i>14.04.2020</i>	
15.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>15.04.2020</i>	
16.	<i>Постановка та формалізація задачі</i>	<i>16.04.2020</i>	
17.	<i>Розробка інформаційного забезпечення</i>	<i>18.04.2020</i>	
18.	<i>Алгоритмізація задачі</i>	<i>20.04.2020</i>	
19.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>23.04.2020</i>	
20.	<i>Розробка програмного забезпечення</i>	<i>05.05.2020</i>	
21.	<i>Налагодження програми</i>	<i>07.05.2020</i>	
22.	<i>Виконання графічних документів</i>	<i>10.05.2020</i>	
23.	<i>Оформлення пояснювальної записки</i>	<i>14.05.2020</i>	
24.	<i>Подання ДП на попередній захист</i>	<i>15.05.2020</i>	
25.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
26.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

6.1 Види випробувань

Вид випробувань узгоджується з представником компанії-замовника до проведення випробувань.

Здача програмного забезпечення відбувається на комп'ютері розробника системи з встановленим програмним забезпеченням, що відповідає вимогам.

Усі програмні продукти, що створюються в рамках даної системи, передаються замовнику у вигляді:

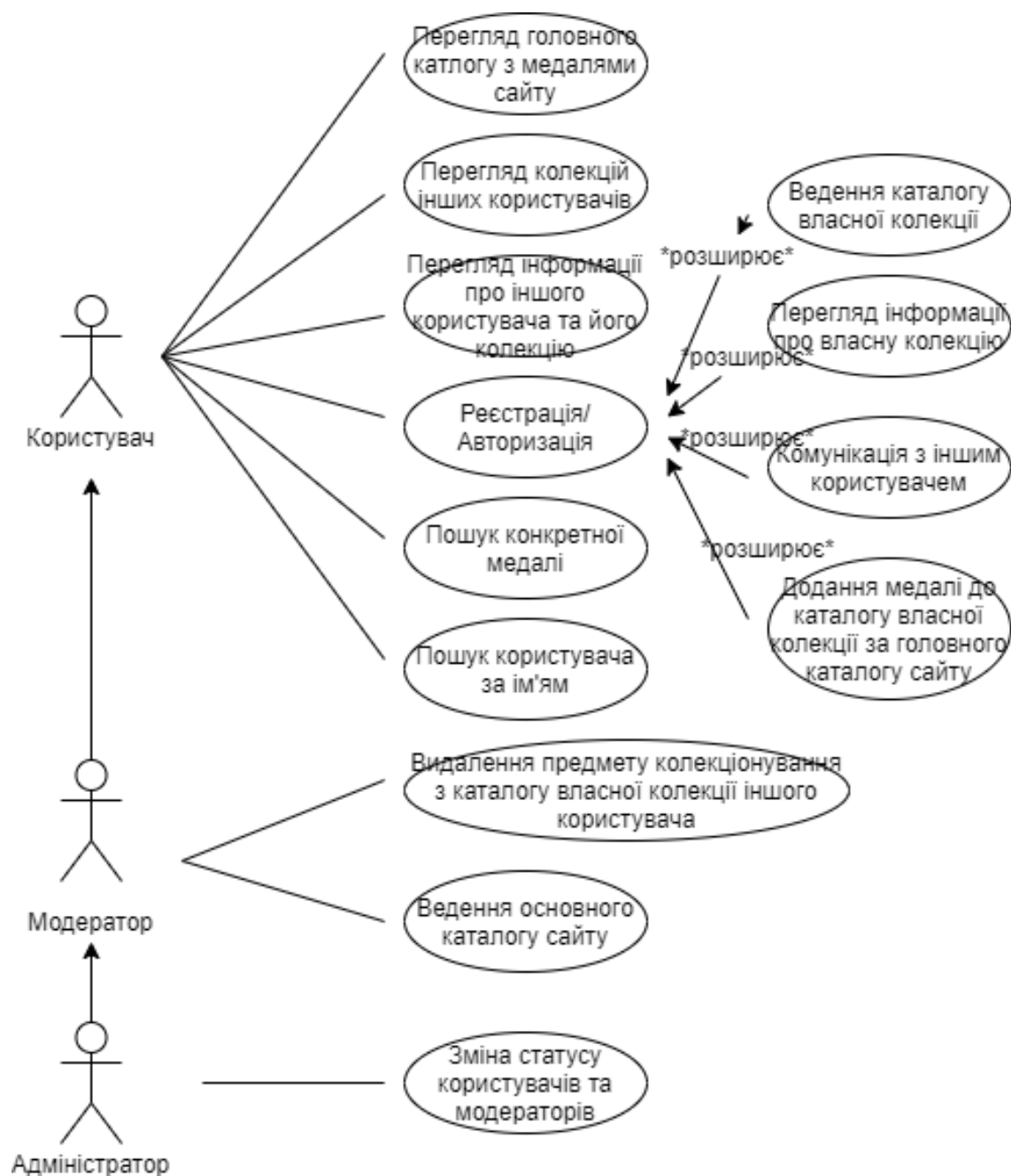
- готових модулів;
- вихідного коду, представленого в електронній формі.

					ДП ІС-6210.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

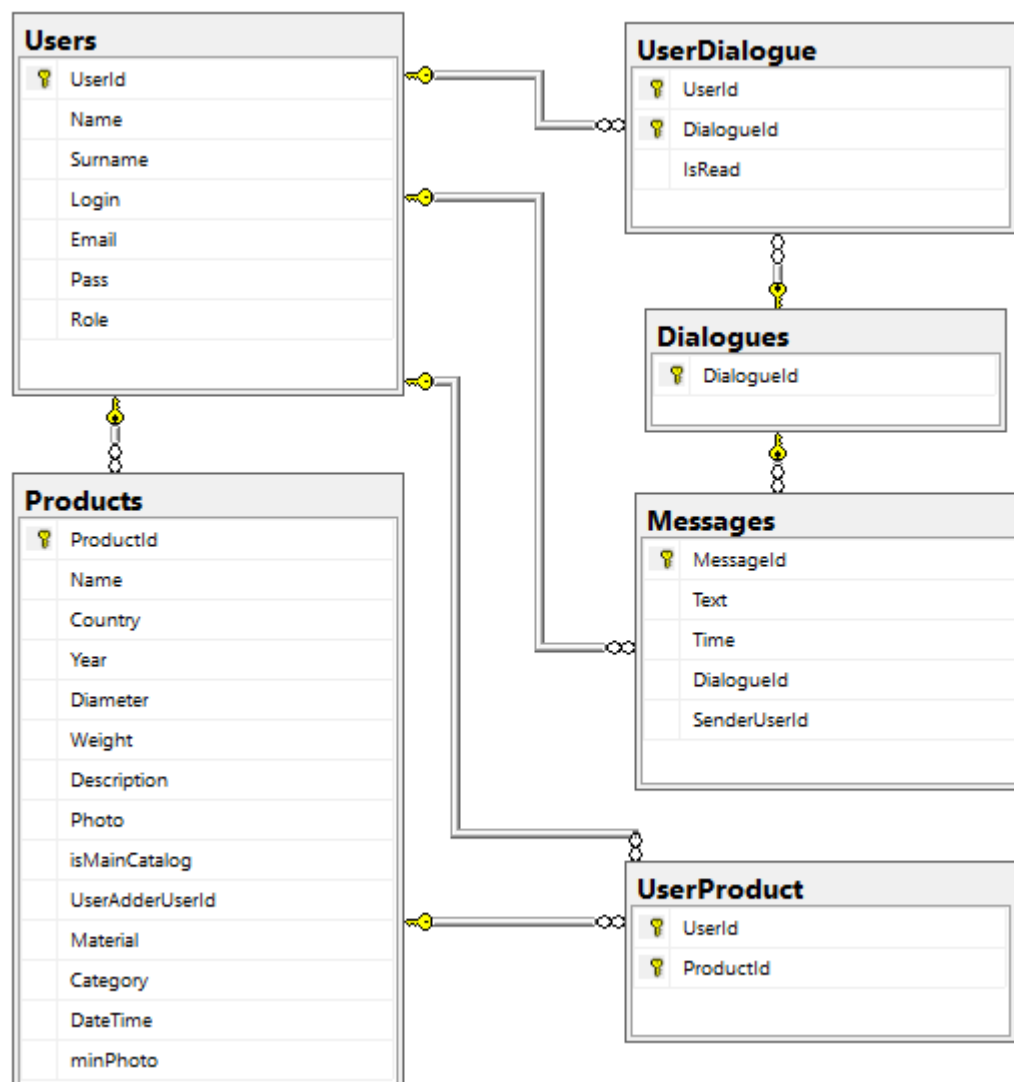
Графічний матеріал до дипломного проєкту

на тему: Інформаційна система підтримки діяльності фалеристів

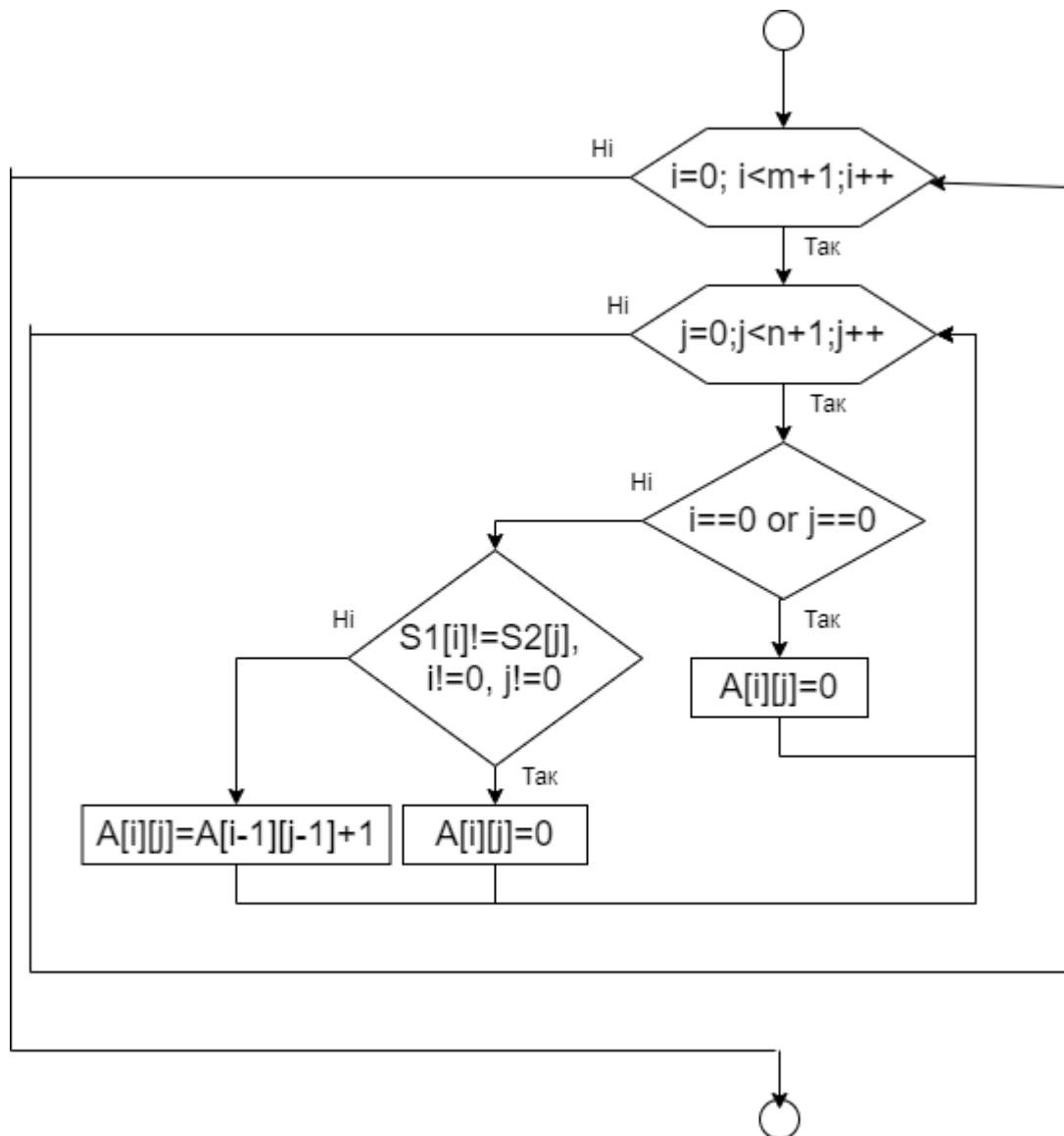
Київ – 2020 року



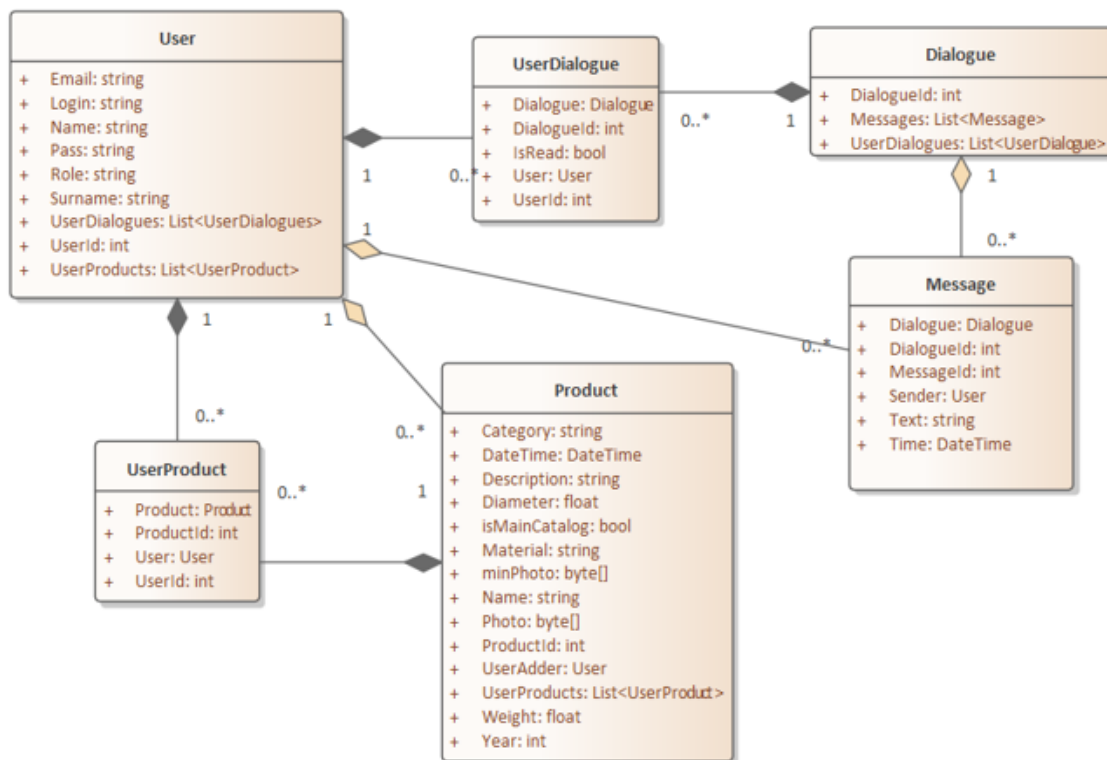
					ДП 6210.02.000 ССВ						
Зм.	Арк.	ПІБ	Підп.	Дата							
Розробн.		Ковинев К.О.			Схема структурна варіантів використання			Літ.	Аркуш	Аркуші	
Керівн.		Жураковська О.С.								1	1
Консульт.								КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62			
Н/контр.		Новінський В.П.									
Затв.		Жураковська О.С.									



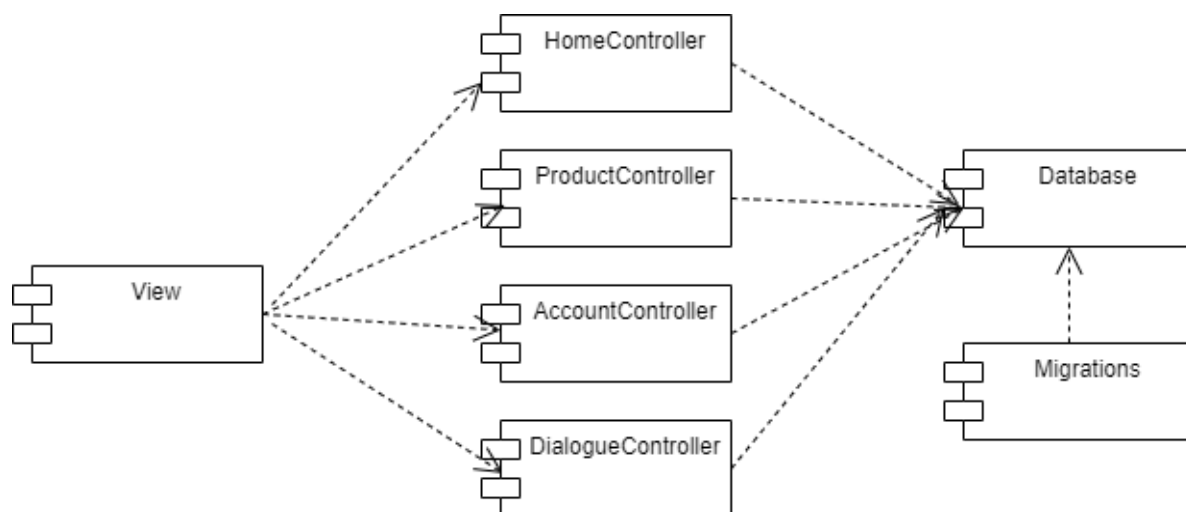
					ДП 6210.03.000 СБД							
Зм.	Арк.	ПІБ	Підп.	Дата								
Розробн.		Ковинев К.О.			Схема бази даних				Літ.	Аркуш	Аркушів	
Керівн.		Жураковська О.С.									1	1
Консульт.												
Н/контр.		Новінський В.П.										
Затв.		Жураковська О.С.										
					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62							



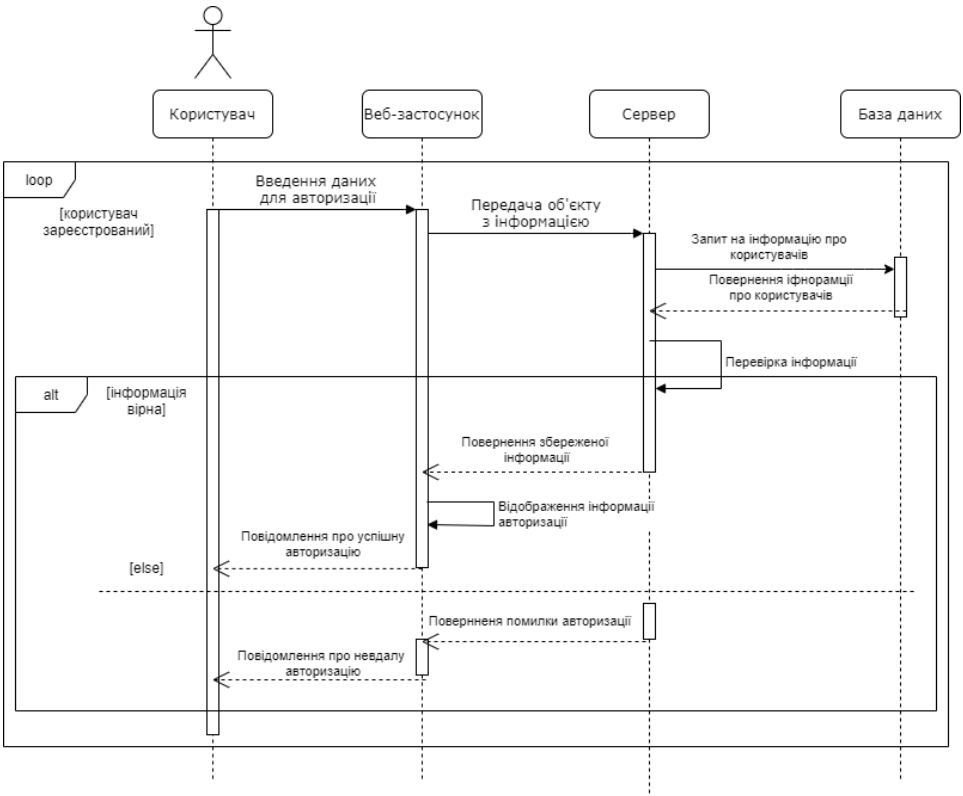
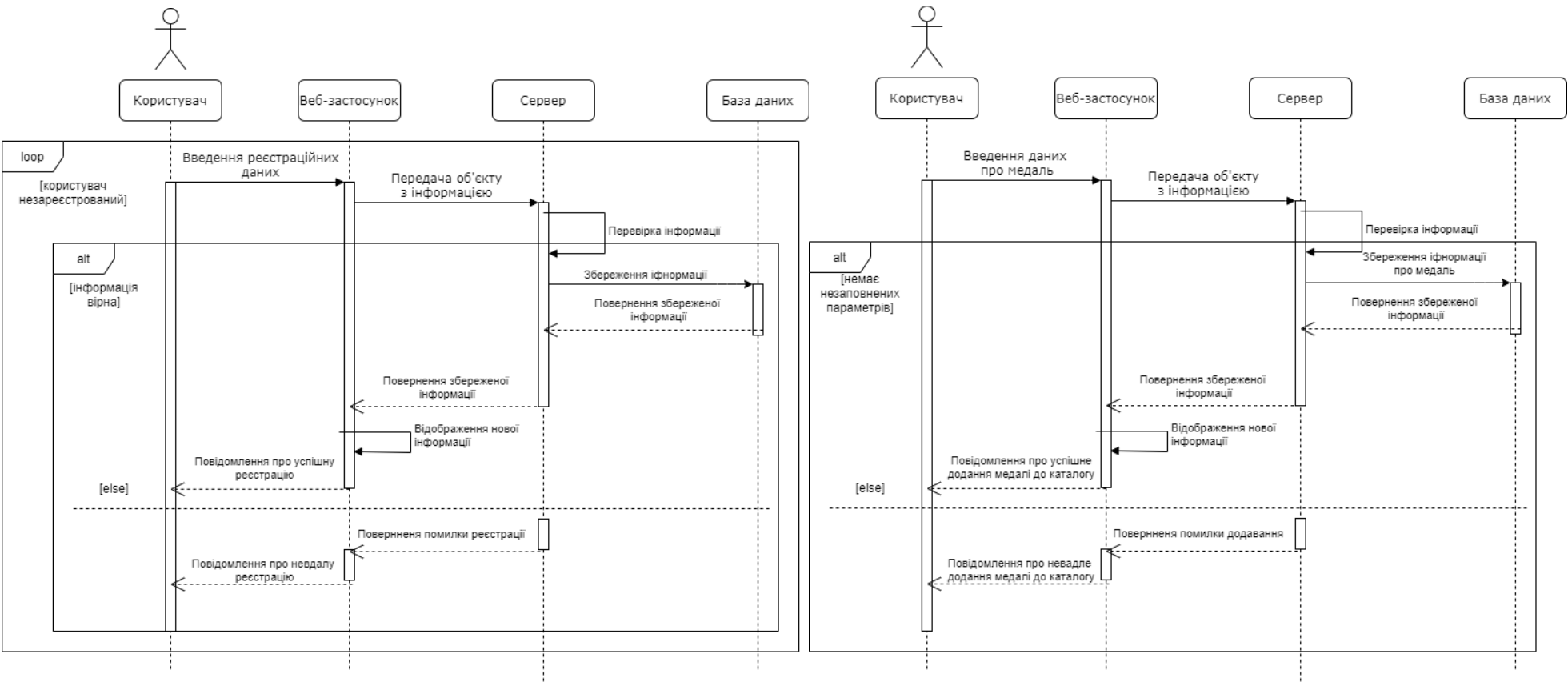
					ДП 6210.04.000 СА		
Зм.	Арк.	ПІБ	Підп.	Дата			
Розробн.		Ковинев К.О.			Схема алгоритму		
Керівн.		Жураковська О.С.					
Консульт.							
Н/контр.		Новінський В.П.					
Затв.		Жураковська О.С.					
						Літ.	Аркуш
						1	1
						КПІ ім. Ігоря Сікорського	
						Каф. АСОІУ	
						Гр. ІС-62	



					ДП 6210.05.000 ССК					
Зм.	Арк.	ПІБ	Підп.	Дата	Схема структурна класів програмного забезпечення			Літ.	Аркуш	Аркушів
Розробн.		Ковинев К.О.								
Керівн.		Жураковська О.С.								
Консульт.										
Н/контр.		Новінський В.П.								
Затв.		Жураковська О.С.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
									1	1



					ДП 6210.06.000 ССК		
Зм.	Арк.	ПІБ	Підп.	Дата			
Розробн.		Ковинев К.О.			Схема структурна компонентів програмного забезпечення	Літ.	Аркуш
Керівн.		Жураковська О.С.					1
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62	
Н/контр.		Новінський В.П.					
Затв.		Жураковська О.С.					
						Аркушів	1



					ДП 6210.07.000 ССП		
Зм.	Арк.	ПІБ	Підп.	Дата	Схема структурна послідовності		
Розробн.	Ковинев К.О.						
Керівн.	Жураковська О.С.						
Консульт.							
Н/контр.	Новінський В.П.						
Затв	Жураковська О.С.						
					Літ.	Аркуш	Аркушів
						1	1
					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		

Вхід на сайт

Введіть Email

Введіть пароль

Увійти

Регістрація

Регістрація

Email:

Ім'я:

Прізвище:

Логін (для відображення на сайті)

Пароль:

Повторіть пароль:

Регістрація

Головний каталог: 7 медалей

Результатів пошуку: 7

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)	Користувач
	Уранова промисловість 35 років	Чеська Республіка	Промисловість	1981	Бронза	60	50	MainAdmin
	Ейнштейн 50 років зі смерті	Чеська Республіка	Наука	2005	Серебро	50	50	MainAdmin
	АЕС Кіншан	Китай	Промисловість	1998	Меднонікель	30	38	MainAdmin
	АЕС Лейбхатл	Швейцарія	Промисловість	1984	Бронза	140	60	MainAdmin
	Ейнштейн 5 франків	Швейцарія	Наука	1979	Меднонікель	30	30	MainAdmin
	Проект Чалк Рівер	Канада	Промисловість	2000	Серебро	20	39	MainAdmin

https://localhost:44301/Home/MyCollection

Пошук користувача:

Знайти

Пошук медалі:

Знайти

Фільтрація:

Країна

Категорія

Рік: Від: До:

Матеріал

Вага: Від: До:

Діаметр: Від: До:

Сортування:

За датою додавання:

Не сортувати

За назвою:

Не сортувати

За країною:

Не сортувати

За категорією:

Не сортувати

Email: kravez95@gmail.com

Логін: petro.kravez

Ім'я: Петро

Прізвище: Кравець

Статус: Користувач







Вийти

https://localhost:44301/Home/MyCollection

					ДП 6210.08.000 КЕ			
Зм.	Арк.	ПІБ	Підп.	Дата	Креслення вигляду екранних форм	Літ.	Аркуш	Аркушів
Розробн.		Ковинев К.О.					1	3
Керівн.		Жураковська О.С.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
Консульт.								
Н/контр.		Новінський В.П.						
Затв.		Жураковська О..С.						

Користувацький каталог: 9 медалей

Результатів пошуку: 9

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)	Користувач
	Роберт Опенгеймер	Франція	Наука	1988	Бронза	130	60	mykola.gyasniy
	АЕС Шуз	Франція	Промисловість	1991	Бронза	120	60	mykola.gyasniy
	АЕС Блас	Франція	Промисловість	1998	Бронза	120	60	mykola.gyasniy
	Марія Кюрі, Великі французь	Франція	Наука	2015	Серебро	20	35	ivan.kazimir
	Інститут Кюрі 50 євро	Франція	Наука	2009	Серебро	60	50	ivan.kazimir
	Марія Кюрі	Франція	Наука	2000	Золото	1	14	ivan.kazimir

https://localhost:44301/Home/MyCollection

Пошук користувача:

Знайти

Пошук медалі:

Знайти

Фільтрація:

Країна

Категорія

Рік: Від: До:

Матеріал

Вага: Від: До:

Діаметр: Від: До:

Сортування:

За датою додавання:

Не сортувати

За назвою:

Не сортувати

За країною:

Не сортувати

За категорією:

Не сортувати



Назва: Уранова промисловість 35 років

Країна: Чеська Республіка










Рік: 1981

Матеріал: Бронза

Моя колекція: 3 медалей

Додати медаль

Результатів пошуку: 3

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)		
	Вільгельм Рентген 150 років	Німеччина	Наука	1996	Серебро	22	35		
	Вільгельм Рентген	Німеччина	Наука	1978	Серебро	31	39		
	Хіросіма	Німеччина	Історія та особистості	2015	Серебро	26	38		

1

https://localhost:44301/Home/MyCollection

Пошук користувача:

Знайти

Пошук медалі:

Знайти

Фільтрація:

Країна

Категорія

Рік: Від: До:

Матеріал

Вага: Від: До:

Діаметр: Від: До:

Сортування:

За датою додавання:

Не сортувати

За назвою:

Не сортувати

За країною:

Не сортувати

За категорією:

Не сортувати




Логін: mykola.gyasniy

Статус: Користувач

Написати повідомлення

Медалей в каталозі: 3

Результатів пошуку: 3

Фото	Назва	Країна	Категорія	Рік	Матеріал	Вага (гр)	Діаметр (мм)	Користувач
	Роберт Опенгеймер	Франція	Наука	1988	Бронза	130	60	mykola.gyasniy
	АЕС Шуз	Франція	Промисловість	1991	Бронза	120	60	mykola.gyasniy
	АЕС Блас	Франція	Промисловість	1998	Бронза	120	60	mykola.gyasniy

1

https://localhost:44301/Home/MyCollection

Пошук користувача:

Знайти

Пошук медалі:

Знайти

Фільтрація:

Країна

Категорія

Рік: Від: До:

Матеріал

Вага: Від: До:

Діаметр: Від: До:

Сортування:

За датою додавання:

Не сортувати

За назвою:

Не сортувати

За країною:

Не сортувати

За категорією:

Не сортувати

					ДП 6210.08.000 КЕ			
Зм.	Арк.	ПІБ	Підп.	Дата	Креслення вигляду екранних форм	Літ.	Аркуш	Аркушів
Розробн.		Ковинев К.О.					2	3
Керівн.		Жураковська О.С.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
Консульт.								
Н/контр.		Новінський В.П.						
Затв.		Жураковська О.С.						

Мої діалоги з:

- mykola.ryasnyy
- ivan.kazimir

Діалог з [ivan.kazimir](#)

14 мая 2020 г.
[petro.kravez](#): Доброго дня! Про яку саме?

14 мая 2020 г.
[ivan.kazimir](#): Привіт. Я хочу дещо дізнатися про медаль Опенгеймера

Відправити

1

https://localhost:44301/Home/MyCollection

Додання медалі:

Назва:

Країна

▼

Категорія

▼

Рік:

Матеріал

▼

Вага (гр):

Діаметр (мм):

Опис:

↑

Виберіть файл

Додати

https://localhost:44301/Home/MyCollection

Зміна:

Назва:

Німеччина

▼

Наука

▼

Рік:

Серебро

▼

Вага (гр):

Діаметр (мм):

Опис:

↑

Вибрати файл

Зберегти

					ДП 6210.08.000 КЕ				
Зм.	Арк.	ПІБ	Підп.	Дата	Креслення вигляду екранних форм	Літ.	Аркуш	Аркушів	
Розробн.		Ковинев К.О.							
Керівн.		Жураковська О.С.					3	3	
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62			
Н/контр.		Новінський В.П.							
Затв.		Жураковська О.С.							